



UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

PROYECTO FIN DE CARRERA

Desarrollo de un Portal Web para la Gestión de Información de Investigación

Alumno: Nieves Pemán Baena

Tutor: Carmen Luisa Padrón Nápoles

Titulación: Ingeniería Superior Informática

DEDICATORIA

A todos los que me han animado y apoyado para que se finalizara y por fin conseguir el primero de los objetivos para el paso al mundo profesional.

Proyecto dedicado a mis padres, mis hermanas, mis abuelos, especialmente mi abuela Cande y mi abuelo Ignacio por haber estado ahí siempre, tanto en verano como en invierno.

Este proyecto va dirigido a Paula, Elena, Rocío, María, Chema y Elisa, esos amigos que siempre están a tu lado.

Proyecto también dirigido para grandes amigos de la universidad como David Virgil, Javier Hernández, Guillermo Suárez, Silvia Rodríguez, Javier de la Blanca, Sergio Pastrana, Jorge Blasco, Javier Paciencia... que han hecho de mi paso por la universidad una experiencia inolvidable.

Índice de Contenidos

1. INTRODUCCIÓN	12
1.1 PROPOSITO DEL DOCUMENTO	12
1.2 OBJETIVOS	13
1.2.1 FUNCIONALIDAD	13
1.2.2 OBJETIVOS DEL AUTOR:	15
1.3 ESTRUCTURA DEL DOCUMENTO:	15
2. ESTADO DE LA CUESTIÓN	17
2.1 ARQUITECTURA DE LA APLICACIÓN WEB	17
2.1.1 ARQUITECTURA MODELO VISTA CONTROLADOR	18
2.1.2 HIBERNATE	20
2.1.3 STRUTS	25
2.1.3.1 FUNCIONAMIENTO DE UNA APLICACIÓN STRUTS	26
2.2 ANÁLISIS COMPARATIVO	29
2.2.1 HIBERNATE, EJB & JDO	29
3. PLANTEAMIENTO DEL PROBLEMA	33
4. SOLUCIÓN	36
4.1 ANÁLISIS DE LOS REQUISITOS.....	36
4.2 IDENTIFICACIÓN DE REQUISITOS	37
4.2.1 REQUISITOS DE CAPACIDAD	40
4.2.2 REQUISITOS DE SEGURIDAD	57
4.2.3 REQUISITOS DE INTERFAZ	58
4.2.4 REQUISITOS DE PORTABILIDAD	66
4.2.5 DIAGRAMA DE CASOS DE USO	67
4.3 DISEÑO.....	71
4.3.1 ARQUITECTURA MVC.....	71
4.3.2 VISTA	72
4.3.3 MODELO: DIAGRAMA ENTIDAD-RELACIÓN (E/R).....	77
4.3.4 CONTROLADOR.....	81
4.4 IMPLEMENTACIÓN DE LA APLICACIÓN.....	82
4.4.1 VISTA	82
4.4.2 CONTROLADOR.....	86
4.4.3 MODELO.....	93
4.4.3.1 PAQUETE HIBERNATE.....	100
4.4.3.2 PAQUETE SERVICIOS.....	102
4.4.3.3 PAQUETE JUNITS	106



4.4.4	DIAGRAMAS DE SECUENCIA	108
5.	GESTIÓN DEL PROYECTO	115
5.1	PLAN DE TRABAJO	115
5.2	DIAGRAMA DE GANTT	117
5.3	ANÁLISIS ECONÓMICO	119
5.3.1	DESGLOSE POR ACTIVIDADES DEL PROYECTO	119
5.3.2	SALARIOS POR CATEGORÍA	120
5.3.3	GASTOS DE PERSONAL IMPUTABLES AL PROYECTO	122
5.3.4	GASTOS DIRECTOS MATERIALES	122
5.3.5	GASTOS INDIRECTOS	123
5.3.6	GASTOS DIRECTOS	124
5.3.7	RESUMEN DEL PRESUPUESTO	125
5.3.8	RESUMEN DEL PRESUPUESTO REAL	126
5.3.8.1	ESFUERZO DEL PROYECTO	126
5.3.8.2	COSTE EFECTIVO DEL PROYECTO	126
6.	CONCLUSIONES.....	128
6.1	CONCLUSIONES	128
6.2	TRABAJOS FUTUROS.....	129
6.3	CONCLUSIONES PERSONALES	131
7.	BIBLIOGRAFÍA.....	132
8.	DEFINICIONES Y ACRÓNIMOS	134
8.1	ACRÓNIMOS.....	134
8.2	DEFINICIONES.....	134
9.	ANEXOS.....	137
A1	ANEXO MANUAL DE USUARIO.....	138
A1.1	INTRODUCCIÓN	139
A1.1.1	LECTORES A QUIENES VA DIRIGIDO	139
A1.1.2	PROPÓSITO	139
A1.2	ACCESO A LA APLICACIÓN.....	143
A1.3	NAVEGAR SIN REGISTRARSE	144
A1.3.1	CAMBIAR DE IDIOMA	144
A1.3.2	VISUALIZAR INFORMACIÓN DE LA AGENDA Y DE LAS NOTICIAS	144
A1.3.3	INFORMACIÓN GENERAL DEL DEPARTAMENTO	145
A1.3.4	MIEMBROS DEL DEPARTAMENTO	146
A1.3.5	DOCENCIA DEL DEPARTAMENTO	148
A1.3.6	OFERTAS DEL DEPARTAMENTO	149
A1.3.7	INVESTIGACIÓN.....	150
A1.3.8	CAMBIAR LA CONTRASEÑA DE USUARIO	152
A1.3.9	VALIDAR USUARIO	153
A1.4	NAVEGAR COMO USUARIO REGISTRADO	154
A1.4.1	VALIDAR USUARIO	154



A1.4.2	VISUALIZAR DATOS DE LA AGENDA Y DE NOTICIAS INTERNAS...	156
A1.4.3	VISUALIZAR DOCUMENTOS: ACTAS O CONSEJOS.....	157
A1.4.4	SUBIR DOCUMENTOS: ACTAS O CONSEJOS.....	157
A1.4.5	VER RESERVAS.....	159
A1.4.6	HACER RESERVAS.....	161
A1.4.7	MODIFICAR RESERVAS.....	164
A1.4.8	ORRAR RESERVAS.....	167
A1.4.9	ALTA USUARIO.....	170
A1.4.10	MODIFICAR USUARIO.....	171
A1.4.11	BAJA USUARIO.....	173
A1.4.12	AÑADIR NOTICIAS.....	174
A1.4.13	MODIFICAR NOTICIAS.....	177
A1.4.14	BORRAR NOTICIAS.....	178
A1.4.15	AÑADIR INFORMACIÓN AGENDA.....	180
A1.4.16	MODIFICAR INFORMACIÓN AGENDA.....	184
A1.4.17	BORRAR INFORMACIÓN AGENDA.....	187
A1.4.18	VOLVER A LA PÁGINA PRINCIPAL DE USUARIOS REGISTRADOS	188
A1.4.19	VOLVER A LA PÁGINA PRINCIPAL DE LA APLICACIÓN.....	189
A1.5	CONFIGURACIÓN E INSTALACIÓN DEL PORTAL WEB.....	190
A2	ANEXO PROPUESTA INICIAL AL CLIENTE.....	197
1.	INTRODUCCIÓN.....	198
1.1.	ALCANCE DEL SOFTWARE.....	198
1.2.	REQUISITOS DE SOFTWARE NECESARIO.....	199
1.3.	DEFINICIONES.....	199
2.	OBJETIVOS DEL SISTEMA A DESARROLLAR.....	201
2.1.	ESTRUCTURA GENERAL.....	201
2.2.	TIPOS DE USUARIOS.....	202
2.3.	PAGINA PRINCIPAL.....	204
2.4.	INFORMACIÓN PÚBLICA.....	206
2.4.1.	¿QUIENES SOMOS?.....	206
2.4.2.	PERSONAL.....	209
2.4.3.	PÁGINA PERSONAL.....	211
2.4.4.	DOCENCIA.....	211
2.4.5.	OFERTAS.....	211
2.4.6.	INVESTIGACIÓN.....	212
2.5.	INFORMACIÓN RESTRINGIDA.....	214



Índice de Figuras

Figura 2-1: Arquitectura.....	17
Figura 2-2: Arquitectura MVC	18
Figura 2-3: Patrón MVC	18
Figura 2-4: Ejemplo Hibernate	21
Figura 2-5: Ejemplo Hibernate (2).....	21
Figura 2-6: Ejemplo Struts	25
Figura 2-7: Ejemplo Struts (2)	26
Figura 2-8: Ejemplo diagrama de clases	28
Figura 2-9: Ejemplo diagrama de colaboración	28
Figura 4-1: Casos de uso-Usuario Público.....	67
Figura 4-2: Casos de uso- Usuario Registrado.....	68
Figura 4-3: Casos de uso-Usuario Administrador.....	69
Figura 4-4: Estructura principal de las páginas.....	72
Figura 4-5: Boceto de página principal.....	73
Figura 4-6: Boceto de página de ¿Quiénes somos?	74
Figura 4-7: Página principal usuarios registrados.....	75
Figura 4-8: Diseño de páginas usuario registrado.....	76
Figura 4-9: Conjunto de JSP's del paquete Vista	84
Figura 4-10: Clases que implementan Struts	87
Figura 4-10: Clases del paquete Modelo.....	95
Figura 4-11: Hibernate.cfg.xml.....	99
Figura 4-12: Paquete Hibernate	100
Figura 4-13: Ejemplo AbstractTipoUsuario	101
Figura 4-14: GestionServicios (1).....	102
Figura 4-15: GestionServicios(2).....	103
Figura 4-16: GestionServicios(3).....	104
Figura 4-17: Paquete JUnit (1).....	106
Figura 4-18: Paquete Junit (2).....	106
Figura 4-19: Paquete JUnit (3).....	107
Figura 4-20: Diagrama de secuencia principal.....	109
Figura 4-21: Diagrama de secuencia de consulta.....	110
Figura 4-22: Diagrama de secuencia de añadir información (1).....	111



Figura 4-23: Idiomas_subir_doc.jsp	112
Figura 4-24: Diagrama de secuencia añadir reserva	114
Figura A1- 1: Página principal.....	141
Figura A1- 2: Logo cabecera	141
Figura A1- 3: Zona de identificación.....	142
Figura A1- 4: Pantalla principal de la aplicación.....	143
Figura A1- 5: Evento.....	144
Figura A1- 6: Noticia	145
Figura A1- 7: Contenido de quiénes somos	145
Figura A1- 8: Menú principal	146
Figura A1- 9: Contenido personal.....	146
Figura A1- 10: Personal PDI.....	147
Figura A1- 11: Menú principal. Docencia	148
Figura A1- 12: Ejemplo licenciatura.....	149
Figura A1- 13: Menú principal. Ofertas.....	150
Figura A1- 14: Contenido investigación.....	151
Figura A1- 15: Ejemplos Grupos de Investigación.....	152
Figura A1- 16: Cambio clave usuario	153
Figura A1- 17: Validación usuario.....	153
Figura A1- 18: Pantalla principal usuario registrado	154
Figura A1- 19: Operaciones del menú de las páginas de Usuario Registrados	155
Figura A1- 20: Pantalla principal usuarios registrados. Agenda	156
Figura A1- 21: Más información	156
Figura A1- 22: Menú de usuarios registrados. Gestión de documentos	157
Figura A1- 23: Menú usuarios registrados. Subir documentos.....	157
Figura A1- 24: Subir documentos. Campos.....	158
Figura A1- 25: Formulario subir documentos.....	159
Figura A1- 26: Menú usuarios registrados. Visualización reservas.....	159
Figura A1- 27: Menú ver reservas	160
Figura A1- 28: Reservas de ordenadores	161
Figura A1- 29: Menú usuarios registrados. Hacer reservas	162
Figura A1- 30: Facilidad de acceso a formulario.....	162



Figura A1- 31: Formulario Reservas	163
Figura A1- 32: Menú modificar reservas	164
Figura A1- 33: Modificar o borrar reservas	165
Figura A1- 34: Modificar reserva de ordenador	166
Figura A1- 35: Mensaje de error (1)	166
Figura A1- 36: Menú borrar o modificar reservas	167
Figura A1- 37: Borrar o Modificar Reservas	168
Figura A1- 38: Borrar Reserva Laboratorio	169
Figura A1- 39: Mensaje de error (2)	169
Figura A1- 40: Mensaje de éxito.....	169
Figura A1- 41: Mensaje de error (3)	170
Figura A1- 42: Campos alta usuario	170
Figura A1- 43: Opción de ayuda en altas usuarios	171
Figura A1- 44: Introducir DNI-1º paso de Modificación Usuario	172
Figura A1- 45: Mensaje de error (4). Del 1º paso Modificación Usuario.....	172
Figura A1- 46: Formulario rellenado-2º paso de Modificación de Usuario ...	173
Figura A1- 47: Operación Baja de usuario	173
Figura A1- 48: Borrar usuario.....	174
Figura A1- 49: Subir noticia	175
Figura A1- 50: Rellenar noticia Idiomas.....	176
Figura A1- 51: Menú modificar noticia	177
Figura A1- 52: Información pública o privada	177
Figura A1- 53: Modificar Noticia	178
Figura A1- 54: Menú borrar noticia	178
Figura A1- 55: Información noticia privada o pública	179
Figura A1- 56: Seleccionar noticia para borrar.....	179
Figura A1- 57: Añadir Evento en la Agenda	180
Figura A1- 58: Añadir información agenda	181
Figura A1- 59: Seleccionar idiomas restantes en añadir evento	182
Figura A1- 60: añadir información de la agenda (Inglés).....	183
Figura A1- 61: Información Agenda Idioma Catalán	183
Figura A1- 62: Modificar agenda.....	184
Figura A1- 63: Información agenda pública o privada	185



Figura A1- 64: Lista de eventos públicos	185
Figura A1- 65: Modificar agenda.....	186
Figura A1- 66: Menú borrar agenda	187
Figura A1- 67: Información agenda pública o privada	187
Figura A1- 68: Borrar noticia.....	188
Figura A1- 69: Volver a la página principal de usuarios registrados.....	188
Figura A1- 70: Opción Desconectar	189



Índice de Tablas

Tabla 8-1: Acrónimos	134
Tabla 5-1: Planificación del proyecto	115
Tabla 5-2: Tareas Gantt.....	118
Tabla 5-3: Diagrama de Gantt-Hitos.....	118
Tabla 5-4. Estimación Tiempo dedicado a las distintas actividades del Proyecto	120
Tabla 5-5. Coste Salarios puesto/hora.....	120
Tabla 5-6. Porcentajes.....	121
Tabla 5-7. Coste salarios de personal/fase	121
Tabla 5-8. Coste horas/empleados	122
Tabla 5-9. Gasto Materiales	122
Tabla 5-10. Gastos Indirectos	123
Tabla 5-11. Gastos Directos.....	124
Tabla 5-12. Presupuesto	125
Tabla 5-13: Coste empleado/hora	126
Tabla 5-14. PresupuestoReal.....	126



1. INTRODUCCI N

“La calidad nunca es un accidente; siempre es el resultado de un esfuerzo de la inteligencia”. John Ruskin

1.1 PROPOSITO DEL DOCUMENTO

En el presente documento presenta un proyecto final de carrera orientado a un trabajo realizado para el mundo empresarial.

El proyecto nace de la petici n del Departamento de Producci n Vegetal y Ciencia de la Universidad de L rida, el cual necesita un gestor de contenidos para satisfacer las necesidades de consulta e inserci n de informaci n p blica y privada del departamento. Se distinguen dos tipos de usuarios: *personal externo*, usuarios ajenos al departamento, y *personal interno* al mismo. Dependiendo del tipo de usuario se realizar n distintas funcionalidades, un usuario *externo* s lo podr  realizar operaciones de consulta p blica y un *personal interno*, aparte de consultar informaci n p blica, podr  consultar informaci n privada del departamento y podr  insertar informaci n p blica y/o privada del departamento.

Para satisfacer sus necesidades se estudiaron sus peticiones y se ha desarrollado un proyecto gen rico, de manera que pueda ser reutilizado por otra universidad u organismo con necesidades similares.

En el desarrollo de esta aplicaci n se ha utilizado la tecnolog a de Struts, una plataforma que permite desarrollar aplicaciones Web en Java bas ndose en el patr n Model-View-Controller (MVC), proporcionando una arquitectura estructurada que divide la l gica de negocio (Model), presentaci n (View) y control de flujo de aplicaciones (Controller).

La aplicaci n hace uso de los paradigmas de la programaci n orientada a objetos y bases de datos relacionales. Existen varias tecnolog as que implementan la persistencia de objetos, simplificando la tarea de conectar bases de datos relacionales y el lenguaje Java, en nuestro caso se ha utilizado Hibernate ya que podr a satisfacer estas necesidades f cilmente, adem s se posee dominio del mismo por asignaturas cursadas



durante la carrera, y sobretodo se contaba con un fácil entorno de desarrollo adaptado para su utilización.

1.2 OBJETIVOS

El objetivo de este proyecto es el Desarrollo de un Portal Web para la Gestión de Información de Investigación de un departamento de una universidad, cubriendo sus necesidades con una solución fiable y fácil de usar y a la vez que sea genérica para que se pueda adaptar a otra universidad. Para ello seguimos el documento de propuesta aprobado por el cliente, el cual se incluye dentro de la sección de anexos. Los objetivos que se deben satisfacer son:

- Las funcionalidades que desea el cliente en el portal Web, las cuales se detallan a continuación.
- Objetivos personales que tiene el autor del mismo.

1.2.1 FUNCIONALIDAD

De modo general se comentan las funcionalidades requeridas por el cliente.

Existirán dos tipos de usuarios del sitio: usuarios *registrados* y usuarios *no registrados*. Se agrupan dentro de los *usuario registrados* aquellos usuarios que pertenecen al departamento de Investigación, para ello se les dará una clave y un usuario para acceder a la información confidencial del departamento, mientras que los *usuarios no registrados*, se corresponden con usuarios externos al departamento, por lo que no tienen usuario ni paswoord y la única operación que pueden realizar es consultar información publicada por el departamento, que es visible para todos los usuarios. Por este motivo, dependiendo del usuario si **es registrado o no registrado**, podrán realizar más o menos operaciones:

❖ Usuario no registrados:

- Los usuarios no registrados del sistema sólo podrán visualizar los resultados de las siguientes consultas:
 - i. Noticias y agenda de la pantalla principal.
 - ii. ¿Quiénes somos?
 - 1. Presentación del departamento.
 - iii. Docencia
 - 1. Docencia de Grado
 - 2. Masters



3. Cursos de especialización

Con el acceso a información complementaria de cada uno de las asignaturas, titulaciones, masters o cursos.

iv. Investigación

1. Grupos de investigación

2. Tesis doctorales

3. Transferencia de tecnología

Con el acceso a información complementaria de cada uno de los grupos, tesis doctorales, empresas que prestan servicios al departamento.

v. Ofertas

1. Becas

2. Proyectos de Final de Carrera

3. Trabajos dirigidos

❖ **Usuario registrado:** Los usuarios registrados deben estar identificados en una Base de Datos como miembros del Departamento o como usuarios administradores. Las operaciones en líneas generales son:

i. Después de identificarse se procede a ver una página que contendrá información propia del departamento:

1. Noticias internas.

2. Información de agenda interna.

ii. Reservas: los miembros del departamento podrán realizar, modificar, borrar y visualizar reservas de coches, laboratorios, ordenadores y cañones.

iii. Noticias: los miembros del departamento podrán añadir, modificar, borrar y visualizar noticias internas o externas del departamento.

iv. Información en la agenda: los miembros del departamento podrán añadir, modificar, borrar y visualizar información a la agenda. Como información interna o externa.

v. Documentos: los miembros del departamento podrán añadir y visualizar documentos, como pueden ser los documentos de actas o los consejos.

vi. Cambiar la clave de acceso a la aplicación.

vii. Modificar sus datos personales.



- ❖ **Usuario registrado como administrador:** Este podrá realizar cualquier funcionalidad comentada anteriormente, pero además podrá realizar las tareas de:
 - i. Usuarios: el administrador se encargará de la gestión de los usuarios, altas de usuarios del departamento, bajas y modificaciones.
 - ii. Reservas: el administrador podrá modificar y borrar cualquier reserva de un usuario.

1.2.2 OBJETIVOS DEL AUTOR:

La autora de este proyecto, además de los objetivos generales, tiene los siguientes retos personales:

- Aplicar los conocimientos adquiridos en la titulación, especialmente los referidos al desarrollo, la ingeniería del software, la programación orientada a objetos y los lenguajes de modelado.
- Mejorar y profundizar las habilidades de trabajo en los entornos de desarrollo, los patrones de diseño y la arquitectura Modelo-Vista-Controlador. Ampliar conocimientos de tecnologías como Struts e Hibernate.
- Realizar el proyecto íntegramente con tecnologías de código abierto, para demostrar alternativas a las existentes en el mercado para el desarrollo de aplicaciones Web multicapa bajo la arquitectura MVC.
- Realizar un proyecto que sea finalmente implantado.
- Conocer las funcionalidades que nos ofrecen los editores de páginas Web, como es el caso del Dreamweaver, y poder realizar diseños más elaborados.

1.3 ESTRUCTURA DEL DOCUMENTO:

Este documento lo componen varios capítulos en los cuales se presenta la siguiente estructura de contenidos:

1. Introducción: se hace un resumen del propósito del documento, comentando los objetivos que se pretenden alcanzar con el desarrollo del portal Web, con las necesidades funcionales que se deben cumplir.
2. Estado de la cuestión: En este capítulo se comenta cual es la situación actual, las tecnologías que se han estudiado para la elaboración del



proyecto. Para ello se hace un análisis comparativo de la arquitectura que mejor se ajusta a nuestra problemática y se explica cuál es su funcionamiento.

3. Planteamiento del problema: En este apartado se hace un análisis detallado de las necesidades que tiene el cliente, las funcionalidades que desea tener en el portal Web.
4. Solución: Después de realizar un análisis de las tecnologías que mejor se ajustan a nuestra problemática e identificar las funciones que el cliente quiere tener en el portal Web se explica cuál es la solución.

Para identificar las funciones a desarrollar se hace un análisis detallado de los *requisitos* del cliente y como consecuencia se identifican los requisitos de capacidad, los de seguridad, los de interfaz y los de portabilidad.

A continuación se explica el *diseño* que satisface estos *requisitos* identificados y por último se detalla cómo se *implementa* esta solución en el portal Web.

Por lo que el apartado de solución constará de:

- a. Identificación de requisitos.
 - b. Diseño.
 - c. Implementación del portal Web.
5. Gestión del proyecto: En este capítulo se comenta el plan de trabajo desempeñado para la realización del proyecto, la planificación de tareas, diagrama de gantt y el análisis económico.
 6. Conclusiones: Para finalizar se comentan las conclusiones obtenidas como resultado de la elaboración del proyecto y se exponen los aspectos que se podrían mejorar como trabajos futuros.
 7. Bibliografía: En este apartado se enumeran las referencias de la literatura utilizada en todo el proyecto.
 8. Definiciones y acrónimos: En este apartado, como el propio nombre indica, se encuentran las definiciones y acrónimos utilizados en el documento.
 9. Anexos: En este apartado se incluyen documentos complementarios como el Manual de usuario y el Documento de propuesta que aprobó el cliente.

2. ESTADO DE LA CUESTIÓN

En este apartado se especifican todas las tecnologías estudiadas para la elaboración del proyecto.

2.1 ARQUITECTURA DE LA APLICACIÓN WEB

Las aplicaciones Web están organizadas siguiendo una arquitectura de tres capas, donde la capa cliente, implementada mediante páginas Web, tiene como misión la captura de los datos de usuario y su envío a la capa intermedia, así como la presentación de resultados procedentes de ésta. Es la capa intermedia la que constituye el verdadero núcleo de la aplicación Web, encargándose del procesamiento de los datos de usuario y de la generación y envío de las respuestas a la capa cliente. Durante este proceso, la capa intermedia deberá interactuar con la capa de datos que se encargará del almacenamiento y recuperación de información manejada por la aplicación.

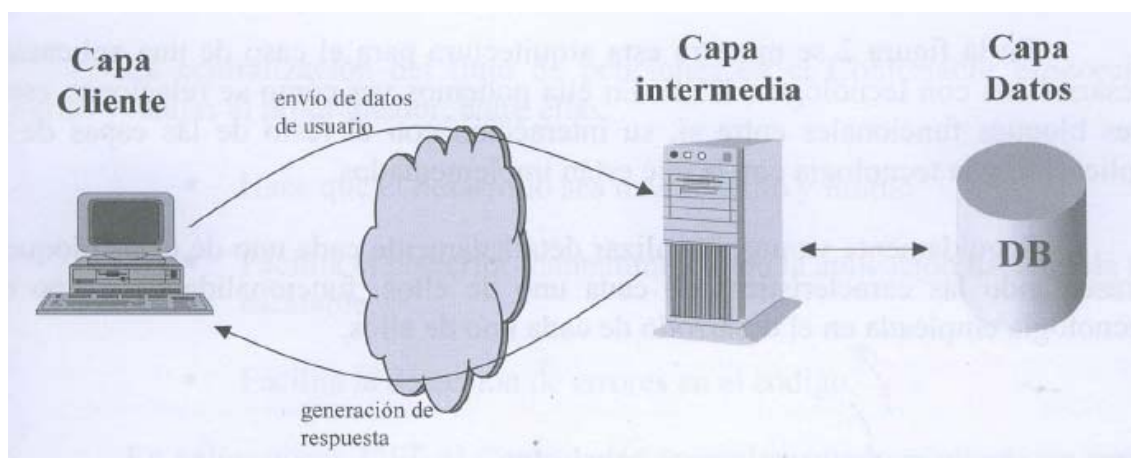


Figura 2-1: Arquitectura

Son muchas las tecnologías y lenguajes que los programadores tienen a su disposición para acometer el desarrollo de la capa intermedia de la aplicación Web (Java/J2EE, PHP, ASP.NET, etc.). No obstante, de cara a afrontar con éxito su implementación, se hace necesario establecer un modelo o esquema que permita estructurar esta capa en una serie de bloques o componentes, de modo que cada uno de estos bloques tenga unas funciones bien definidas dentro de la aplicación y pueda desarrollarse de manera independiente al resto [1].

Uno de estos esquemas y el más utilizado por los desarrolladores J2EE, es la arquitectura Modelo Vista Controlador (MVC), la cual proporciona una clara separación entre las distintas responsabilidades de la aplicación.

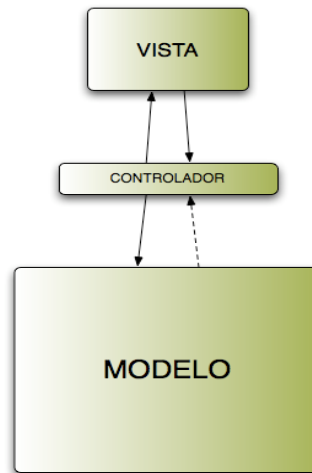


Figura 2-2: Arquitectura MVC

2.1.1 ARQUITECTURA MODELO VISTA CONTROLADOR

El patrón MVC (Modelo, Vista, Controlador). El cual es un patrón Software aportado originalmente por el lenguaje SmallTalk a la Ingeniería del Software, su uso más extendido es en el desarrollo de aplicaciones Web.

El principal objetivo de este patrón de diseño es separar la forma de acceder a los datos (Modelo), de la presentación de los datos para el usuario (Vista), así como de los procesos que realizan la lógica de negocio de la aplicación (controlador) [1]:

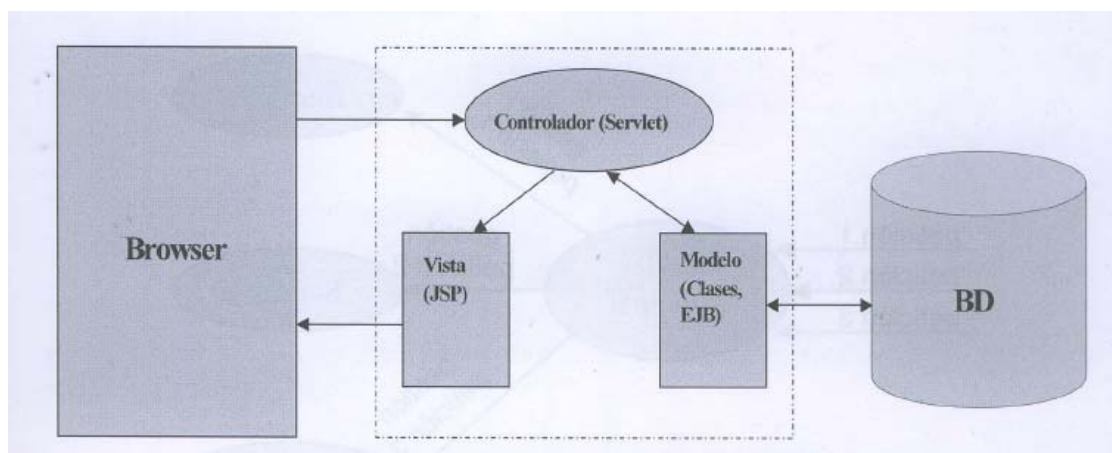


Figura 2-3: Patrón MVC



- **Modelo:** Contiene el núcleo de la funcionalidad de la aplicación, encapsula el estado de la misma, la lógica de negocio de la aplicación, el acceso a los datos y su manipulación y es independiente del controlador y la vista. Para obtener los elementos de la base de datos, hemos utilizado Hibernate, con el que se consigue realizar un *Mapeo objeto-relacional* para la plataforma Java, de esta manera trabajamos directamente con objetos JAVA que son los asociados a las tablas del modelo de la Base de Datos. No obstante, más adelante detallaremos la tecnología en sí [1] [2].
- **Vista:** Es la encargada de la presentación del modelo, pudiendo acceder al modelo pero nunca cambiando su estado y pudiendo ser notificada cuando hay un cambio en el estado del modelo. Para realizar la presentación de los datos se utilizan las páginas **JSP** (Java Server Pages) [1] [2].
- **Controlador:** Reacciona a las peticiones que realiza el cliente, ejecutando para cada una de ellas las acciones adecuadas y creando el modelo pertinente. Para representar el papel de controlador se pueden utilizar los *servlets*, los cuales pueden ser invocados desde páginas JSP o desde otros *servlets* con unos parámetros de entrada (datos de un formulario...), para realizar las acciones (llamadas al modelo u otras operaciones) y redirigir los resultados a una página (HTML o JSP). Para esta redirección es óptima la utilización de **STRUTS** o **JSF**, ya que permiten tener en un fichero todas las posibles redirecciones en función del resultado del *servlet*. Y en nuestro caso hemos optado por la utilización **STRUTS** [1] [2].



2.1.2 HIBERNATE

El objetivo de este apartado es familiarizar al lector con la filosofía utilizada en Hibernate.

Hibernate es una herramienta de mapeo objeto-relacional para la plataforma Java, que tiene como objetivo facilitar la persistencia de objetos Java en bases de datos relacionales y al mismo tiempo la consulta de estas bases de datos para obtener objetos [5].

Las posibilidades que ofrece Hibernate exceden con mucho las pretensiones de esta explicación, así que me centraré sólo en las que cubren la casuística más habitual.

Cualquier persona que se dedique al desarrollo orientado a objetos se encontrará con la problemática de implementar una persistencia de objetos en un soporte relacional.

Muchas veces cuando se diseña no se tiene un modelo único, sino que se hace un modelo orientado a objetos de manera independiente al modelo de datos. El problema es que hay un *divorcio* total entre la estructura relacional y la orientada a objetos. Para explicarlo de la manera más breve, pondré un ejemplo de diseñar una base de datos para una escuela [9].

Por una parte, tendríamos la lista de personas que pertenecen a la escuela.

- Cada persona tiene una sola categoría: profesor, alumno, personal de administración, etc.
- En la escuela, hay niveles: primero de EGB, segundo de EGB, etc.
- Además, hay clases. Cada clase pertenece a un nivel y tiene un nombre. Así, es posible que haya tres clases de primero de EGB: A, B y C.
- Pero una clase no es una clase sin alumnos y profesores. Así que tendremos que asignar personas a una clase.

No entraremos aquí en un diseño minucioso ni nos preocupará si una persona está en una o más clases.

El siguiente gráfico muestra las relaciones entre tablas:

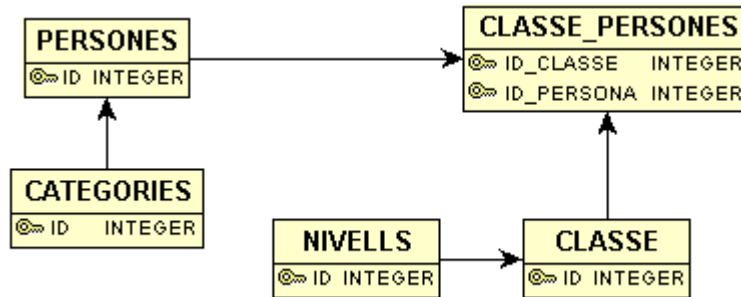


Figura 2-4: Ejemplo Hibernate

Pero el diseño orientado a objetos con el que nos gustaría trabajar no es un calco del diseño relacional. La siguiente figura ilustra nuestro diseño orientado a objetos con un diagrama de clases UML:

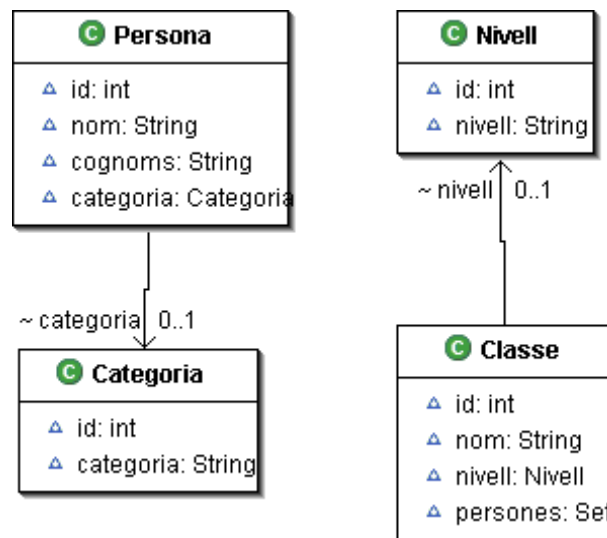


Figura 2-5: Ejemplo Hibernate (2)

Observemos algunos cambios en nuestro diagrama de clases:

- Los nombres de los objetos están en singular, mientras que los de las tablas están en plural. Así, entendemos que una instancia de un objeto se corresponde con un registro de una tabla. Una tabla es un conjunto de Personas. La clase Persona representa una sola persona.
- La clase Persona no tiene un entero que apunte a la tabla Categorías. Tiene un objeto de tipo Categoría. Así, vemos que los tipos de las propiedades de un objeto no siempre se corresponden con los tipos de las columnas de las tablas. Un caso similar, lo encontraríamos en el objeto Clase.



La propiedad correspondiente a la columna `ID_NIVEL` `INTEGER` es del tipo `Nivel`, no `int`.

- En nuestro diseño del modelo relacional, tenemos la tabla `CLASE_PERSONAS` que es la encargada de almacenar las personas asociadas a cada clase. Desde el punto de vista orientado a objetos, el planteamiento es distinto. Cada clase tiene un conjunto, `Set`, de personas. No necesitamos crear un objeto puente que relacione clases y personas.
- **Conclusión: el modelo relacional y el orientado a objetos no acaban de “casar”.** Esto se debe a que el diseño se ha realizado de manera independiente, pero si hubiera un diseño único del modelo de datos, deberían de ser iguales. Por lo cual la solución es la utilización de una herramienta que nos facilite el mapeo de los dos modelos.

Ahora bien, está claro que tendremos que hacer “casar” los dos modelos si queremos trabajar con comodidad; y este es el papel de Hibernate: hacer de *casamentero*. Hibernate se encarga de casar los dos modelos de manera que nosotros trabajemos desde Java como corresponde.

Usando el modelo de objetos:

- **Sin Hibernate**, para añadir un registro a la tabla `Categorias` tendríamos que escribir algo similar a esto:

```
[...]
Class.forName("org.hsqldb.jdbcDriver");
String url = "jdbc:hsqldb:./Databases/ESCUELA";
Connection connection = DriverManager.getConnection(url, "sa", "");
String ins = "INSERT INTO CATEGORIAS VALUES(NULL, 'Profesor')";
Statement stmt = null;
stmt = connection.createStatement();
stmt.executeUpdate(ins);
[...]
```

- **Con Hibernate**, escribiríamos algo similar a esto:

```
[...]
Configuration conf = new Configuration();
conf.addClass(Categoria.class);
SessionFactory sessionFactory = conf.buildSessionFactory();
Session session = sessionFactory.openSession();
```



```
Categoria categ = new Categoria();
categ.setCategoria("Profesor");
session.save(categ);
session.flush();
session.close();
[...]
```

Y si nos interesase **añadir** otra categoría:

```
[...]
session = sessionFactory.openSession();
categ = new Categoria();
categ.setCategoria("Alumno");
session.save(categ);
session.flush();
session.close();
[...]
```

A partir de estas demostraciones, podemos responder a las siguientes preguntas que se nos plantean:

¿Cuál es la gran diferencia entre el código tradicional JDBC y el código que escribimos con Hibernate?

“Ha desaparecido el SQL”. No se necesita saber SQL ya que es transparente por las operaciones que ofrece Hibernate.

Si lo que queremos es ejecutar una consulta a la base de datos, tendremos que utilizar un lenguaje de interrogación que “case” con el mundo orientado a objetos. Pedir una lista ordenada de profesores nos costaría tan poco como esto:

```
[...]
session = sessionFactory.openSession();
String qry = "FROM escuela.beans.Persona AS P
WHERE P.categoria.categoria = 'Profesor'
ORDER BY P.apellidos, P.nombre";
List profes = session.find(qry);
session.close();
for (int i = 0; i < profes.size(); i++) {
    Persona p = (Persona)profes.get(i);
    System.out.println(p.getNombre() + " " + p.getApellidos() +
    p.getCategoria().getCategoria());
}
```



Observamos aquí que la consulta se formula en base a objetos y que, una vez más, “ha desaparecido el SQL”. Del SQL, se ocupa Hibernate, el cual, casa nuestro mundo objectual con el relacional. Pedimos objetos y expresamos las condiciones de la consulta orientado a objetos. Hibernate se encarga de transformar nuestra consulta al dialecto SQL que toque.

Pero para poder utilizar Hibernate, necesitamos saber:

- ¿Qué gestor de bases de datos está detrás?
- ¿A qué base de datos me conecto?
- ¿Cómo me conecto?
- ¿Cómo se emparejan propiedades y campos de tablas?:
 - Clave primaria:
 - ¿Cuál es la propiedad que se corresponde con la clave primaria de la tabla correspondiente?
 - ¿Qué método deberé utilizar para generar un nuevo valor de la clave primaria?

Una vez conocidas las respuestas a estas preguntas, esta información será plasmada en dos archivos distintos:

- El archivo **de propiedades de Hibernate (Hibernate.properties)** es el encargado de determinar los aspectos relacionados con el gestor de bases de datos y las conexiones con él. Por lo tanto, en él se dice qué gestor de bases de datos usaremos y a qué base de datos nos conectaremos y cómo lo haremos.
- **Los archivos que definen el emparejamiento o correspondencia (mapping)** de propiedades de los objetos (clases) con tablas y columnas (*.hbm.xml), existirá un fichero de éstos por cada tabla del modelo relacional. Los archivos tienen el formato **NombreDeClase.hbm.xml**, en cada uno de ellos se describe cómo se relacionan clases-tablas y propiedades-columnas.

Con todo lo comentado anteriormente, se puede tener una idea de la tecnología. Retomaremos el tema en el apartado *Diseño e Implementación de la aplicación*, explicando cómo se ha utilizado la tecnología en la solución.

2.1.3 STRUTS

Struts es un framework Java para aplicaciones Web que implementa el modelo MVC. Este framework dispone de un núcleo básico (conjuntos de componentes) que permiten implementar la lógica de negocio y separar la lógica de negocio de la presentación.

Struts provee un conjunto de clases y de librerías de *tags* que conforman el Controlador, la integración con el Modelo y facilitan la construcción de vistas, facilitando de este modo el desarrollo de aplicaciones consistentes y bien estructuradas [1] [2].

En una aplicación que hace uso de Struts las acciones que invoca deben estar especificadas: una acción se corresponderá con una clase Java (Heredera de la clase Action Struts). El mapeo de acciones y clases se especifica en un fichero de importancia vital: **struts-config.xml**. Ahí se especifican todas las relaciones entre acciones y clases, formularios y clases, acciones y *jsp*'s de presentación, que globalmente conforman el “mapa” de la aplicación.

Para facilitar la explicación, vamos a utilizar un ejemplo sencillo: *Una página con un saludo*. Se deberían realizar los siguientes pasos:

1. Crear un página JSP (la presentación)
2. Crear una clase Action (componente del controlador)
3. Definir correctamente la clase *Acción* en el **struts-config.xml**

El diagrama de clases correspondiente, que representa el conjunto de clases utilizadas y sus relaciones, sería el siguiente:

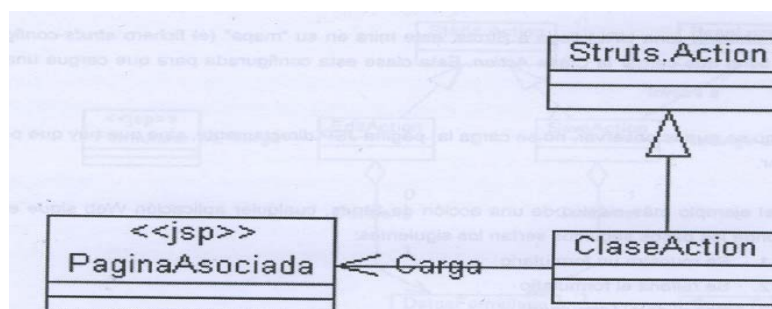


Figura 2-6: Ejemplo Struts

Como se puede ver, no existirían más que dos ficheros, una clase que hereda y extiende de *struts.Action* y una página JSP con la presentación. La clase Action se configura en el **struts-config.xml** y se convierte en parte del controlador.

2.1.3.1 FUNCIONAMIENTO DE UNA APLICACIÓN STRUTS

Cada vez que desde el navegador cliente llega una petición al contenedor tendrá lugar el proceso que describimos a continuación [1]:

- **Análisis de URL** de la aplicación. El contenedor Web pasa la petición al objeto `ActionServlet`, éste, utiliza la última parte de la url de la petición para determinar la acción a realizar.
- **Determinación de la acción a realizar.** Utilizando el dato obtenido anteriormente, el objeto `ActionServlet` realiza una consulta en el **archivo `struts-config.xml`** para determinar las operaciones a realizar. Tras esta consulta `ActionServlet` realiza dos acciones:
 - Crea u obtiene la instancia del `ActionForm` y lo rellena con los datos del formulario cliente.
 - Crea una instancia del objeto `Action` correspondiente e invoca a su método `execute()`, pasándole como argumento una referencia al objeto `ActionForm`.
- **Procesamiento de la petición.** En el método `execute()` de la subclase `Action` se codificarán las acciones para el procesamiento de la petición.
- **Generación de la vista.** Es el resultado del método `execute()`. Muestra la página resultado.

Esta manera de interactuar entre las clases y Struts se representa en el siguiente diagrama de colaboración [2]:

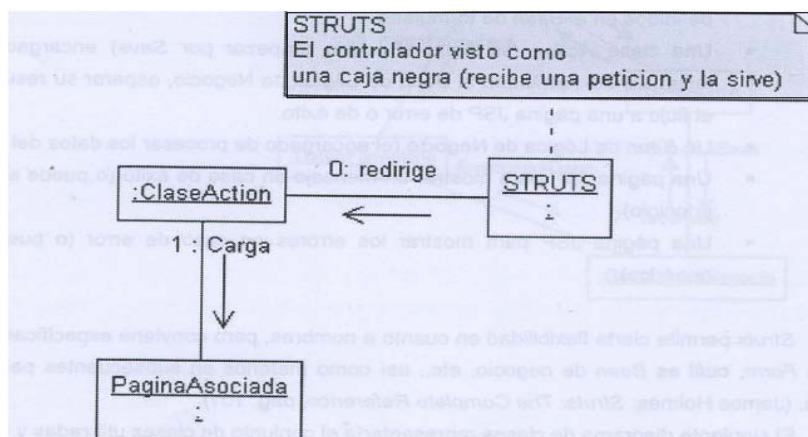


Figura 2-7: Ejemplo Struts (2)



Entonces, una petición (url) llega a Struts; este mira en su “mapa” (el fichero `struts-config.xml`), y deduce que tiene que cargar la Clase Action. Esta clase esta configurada para que cargue una página JSP.

Como se puede observar, no se carga la página JSP directamente, sino que hay que pasar por el controlador.

En el ejemplo más clásico de una acción Struts, cualquier aplicación Web sigue el mismo esquema, donde las acciones serían los siguientes:

1. Se muestra el formulario.
2. Se rellena.
3. Se valida el formulario.
4. El controlador realiza las gestiones oportunas para generar un resultado.
5. Se muestra el resultado.

Todo esto se podría hacer con dos JSP's, pero estaría mezclada la lógica de negocio con la presentación de la aplicación. Así que si utilizamos Struts necesitaríamos lo siguiente:

- Una clase Action (`NombredelaClaseAction.java`) encargada de cargar el formulario.
- Un Bean de tipo *Form* cuyos campos o atributos de instancia coinciden con los del formulario (al ser de tipo *JavaBean* tendrá los correspondientes métodos *set* y *get* para los atributos). Este *form* y sus atributos son requisitos imprescindibles para manejar la información que introduce el usuario en el formulario.
- Una página JSP que contiene el formulario citado. Los campos deben coincidir con los definidos en el Bean de formulario.
- Una clase Action que se encargue de pasar la instancia del formulario al Bean de Lógica de Negocio, esperar su resultado y redirigir el flujo a una página JSP de error o de éxito.
- Un Bean de Lógica de Negocio (encargado de procesar la información recuperada del formulario)
- Una página JSP para mostrar mensajes de error en caso de que se produzca algún error.
- Una página JSP para mostrar mensajes de éxito, cuando es posible generar algún resultado.

El siguiente diagrama de clases representaría el conjunto de clases utilizadas y sus relaciones.

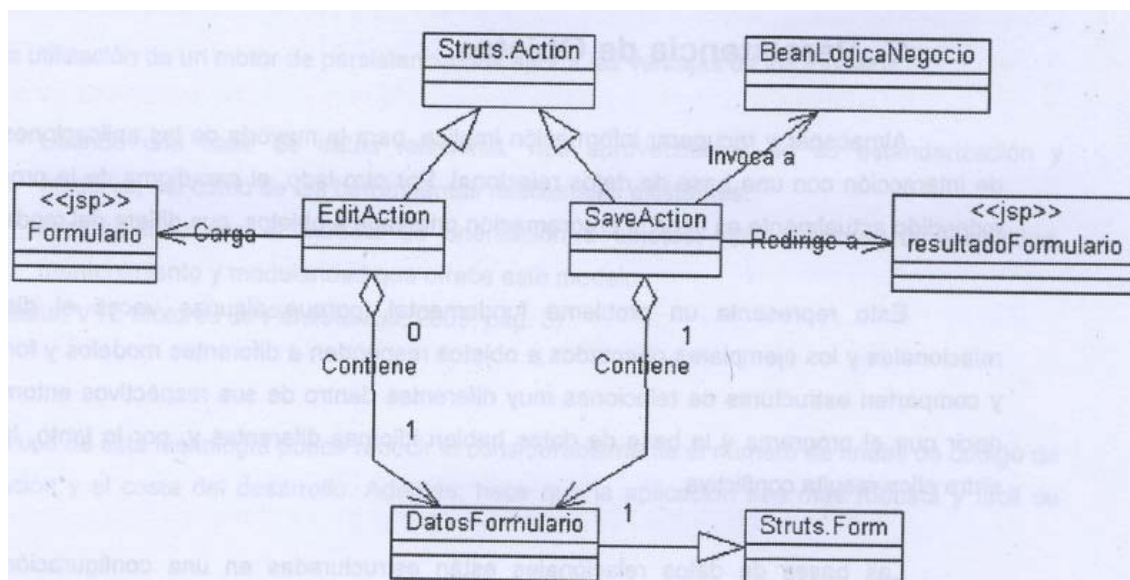


Figura 2-8: Ejemplo diagrama de clases

La manera de interactuar entre las clases se muestra el siguiente diagrama de colaboración.

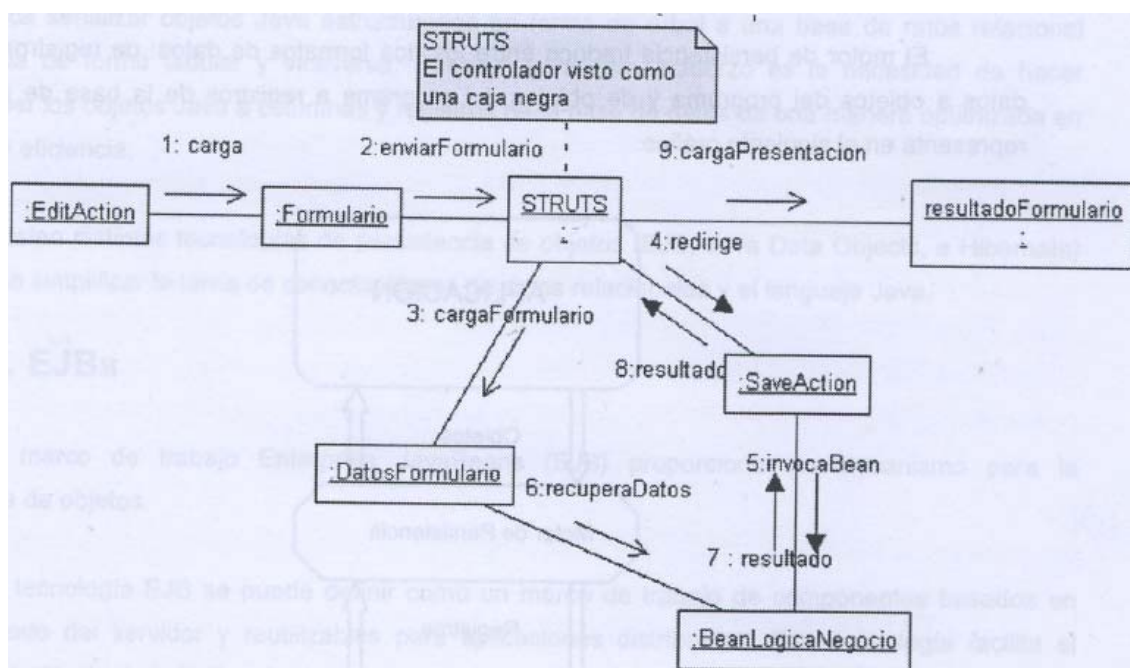


Figura 2-9: Ejemplo diagrama de colaboración



2.2 ANÁLISIS COMPARATIVO

Para la elección de un motor de persistencia de datos, se realizó un estudio comparativo de tres sistemas de persistencia de datos en las bases de datos relacionales. Se comentan las ventajas y desventajas de utilizarlos. La autora del proyecto se decantó por *Hibernate* ya que lo había utilizado en alguna práctica universitaria y poseía mayor dominio que los restantes. Otro motivo es que para el entorno de desarrollo utilizado, *Eclipse*, tiene unas opciones que facilitan el desarrollo de *Hibernate*.

2.2.1 HIBERNATE, EJB & JDO

❖ JDO(Java Data Objects) [10]

○ Ventajas

- Capacidad de trabajar con fuentes de datos no relacionales, por ejemplo con ficheros planos XML, manteniendo una misma interfaz.
- El Programador se abstrae del lenguaje SQL: lenguaje JDOQL, este lenguaje tiene sintaxis Java, es simple, pero al mismo tiempo, una consulta sirve para acceder a sistemas radicalmente diferentes, RDBMS, OODBMS, ficheros o IES.
- Cualquier sistema de persistencia tiene que hacer uso del ciclo de vida de JDO o un subconjunto de éste (siempre que el sistema transaccional no esté soportado). JDO es una especificación que estandariza la forma de programar en Java con algunos motores de persistencia, como son TopLink, Cocobase y FastObjects.

○ Desventajas

- Para que una clase sea persistente, se debe de pasar la clase compilada por el procesador de código Java (Enhancer) para que la habilite en dicha finalidad. Por ejemplo hay que tener cuidado cuando se usa, por primera vez, una clase que representa la clave primaria debemos de asegurarnos que esta compilada y registrada como persistente.



- Para borrar los objetos persistentes, previamente se han de recuperar para después eliminarlos de la Base de Datos. No tiene métodos en las interfaces que faciliten el borrado directo en la Base de Datos sin recuperación previa.
- El lenguaje JDOQL es muy limitado, ya que sólo se puede consultar por clases concretas y no se puede consultar directamente por las interfaces, este hecho perjudica a los programas que pretenden crear código genérico.

❖ **Hibernate** [5]

○ Ventajas

- Herramienta libre bajo licencia LGPL. Es una herramienta madura, creada en el 2001, y es una de las más extendidas.
- Existe la actualidad disponibilidad para su integración en entornos de desarrollo, en Eclipse, se integra a través de pluggins, de esta manera se consigue ayudar a la utilización y automatización de la persistencia. Estos pluggins están dentro del proyecto hibernate-tools.
- Hibernate soporta paradigmas de la orientación a objetos como son el polimorfismo y las asociaciones.
- Ofrece unos métodos de consulta basados en un lenguaje de consultas de datos llamado HQL (Hibernate Query Language). Por lo que no es necesario saber SQL.
- Se pueden crear filtros definidos por el usuario.
- La sesión explícita hace que el flujo de control sea más limpio debido a que sigue el modelo Connection, el cual facilita las transacciones a la Base de Datos.
- No necesita tener una clase que se encargue de la gestión de las conexiones a la Base de Datos ya que Hibernate lo tiene configurado en un fichero *hibernate.cfg.xml*.



○ Desventajas

- El API de las sentencias SQL es un poco complicado, ya que existen varios métodos para las consultas y necesitas conocerlo bien.
- Se necesita conocer el framework Collection de Java para la gestión de las operaciones de consultas.
- Necesitas un entorno de desarrollo que facilite su incorporación ya que sino es bastante costoso generarlo manualmente (la generación de los ficheros correspondientes a las NombreClase.hbm.xml y sus clases Java correspondientes, más la generación de los ficheros de configuración de Hibernate).

❖ **EJB(Enterprise Java Beans)/CMP (Gestionados por el contenedor)**

○ Ventajas

- Portabilidad de la aplicación. Una aplicación EJB puede ser desplegada en cualquier servidor de aplicaciones que soporte J2EE.
- Integración con sistemas no-Java. Las APIs relacionadas, como las especificaciones Connector y Java Message Service (JMS), así como los beans manejados por mensajes, hacen posible la integración de los enterprise beans con sistemas no Java, como sistemas ERP o aplicaciones mainframes.
- El hecho de que la especificación EJB sea un estándar hace que exista una creciente oferta de herramientas y formación que facilita el trabajo del desarrollador de aplicaciones EJB.

○ Desventajas

- EJB (“Enterprise JavaBeans”), incorporan un motor de persistencia a través del mecanismo conocido como “entity beans”. Sin embargo, los “entity beans” no son un mecanismo de persistencia totalmente recomendable, pues no permiten implementar algunas propiedades de la programación orientada a objetos, por ejemplo, herencia y cuando se pretende implementar esta propiedad, genera una complejidad extra.



- CMP (Container Manage Persistente) no pertenece a la especificación EJB. Se ha forzado para que siga un modelo para este propósito.



3. PLANTEAMIENTO DEL PROBLEMA

La Universidad de L rida nos plantea la necesidad de hacer un gestor de contenidos para un Departamento de investigaci n.

Las funciones b sicas que solicitan son consultas, modificaci n, inserci n y borrado de informaci n que se almacena en una Base de Datos propia del departamento. Para abordar la petici n, se realiza una reuni n con el cliente, la directora del proyecto y la autora del mismo y se definen las funcionalidades b sicas a cubrir. Como consecuencia de la reuni n, se realiza un documento de propuesta para el cliente, en este caso para el departamento en cuesti n, con lo que se pretende determinar el conjunto de requisitos del sistema a desarrollar y la conformidad por parte del cliente.

Adem s, nos comunica que aparte de las funcionalidades b sicas, desea que se hagan distinciones entre el tipo de operaciones que puede realizar un usuario del resto, es decir, hay distintas funcionalidades en base a si es un usuario “p blico” o “registrado”, definiendo el t rmino “registrado” como el personal que es miembro del departamento y que est  registrado en la aplicaci n. Resumiendo en l neas generales, las funcionalidades que se acuerdan son las siguientes:

1. Operaciones que se realizan sobre la Base de datos, disponible tanto para usuarios *p blicos* como *registrados*:
 - Consultar informaci n en la Base de Datos:
 - a. Informaci n del personal de la universidad: PDI, PAS, Becarios.
 - b. Informaci n de la docencia que imparte el departamento: Titulaciones, Masteres y cursos.
 - c. Informaci n de Ofertas que ofrece: PFC's (Proyectos Final de Carrera), Becas y Trabajos dirigidos.
 - d. Informaci n de Investigaci n: grupos de investigaci n, tesis y transferencia tecnol gica.
 - e. Documentos: actas, consejos.



- f. Información de Reservas: coches, ordenadores, laboratorios y cañones.
 - g. Información de Noticias y agenda.
 - h. Validar usuario del departamento. Al usuario que no esté registrado le dará un error de validación.
- Añadir información en la Base de Datos relacionado con los siguientes conceptos, disponible sólo para usuarios *registrados*:
 - a. Reservas: de coches, cañones de visualización, laboratorio y ordenador.
 - b. Documentos del departamento: actas o consejos.
 - c. Usuarios Lo puede realizar usuario *registrado* como *administrador*.
 - d. Información de Noticias y de agenda del departamento.
 - e. Cambiar contraseña de los usuarios registrados.
 - Modificar información en la Base de Datos relacionado con los siguientes conceptos, disponible sólo para usuarios *registrados*:
 - a. Reservas: de coche, cañón, laboratorio y ordenador. El usuario *administrador* puede modificar cualquier reserva, independientemente de si la ha realizado él mismo. Los demás sólo pueden modificar información de las reservas realizadas por ellos mismos.
 - b. Información propia del usuario. El *administrador* puede modificar cualquier usuario. Los demás usuarios sólo pueden modificar su información.
 - c. Noticias y agenda.
 - Borrar información en la Base de Datos relacionado con los siguientes conceptos, disponible sólo para usuarios *registrados*:
 - a. Reservas: de coche, cañón, laboratorio y ordenador. El *administrador* puede borrar cualquiera, independientemente de si la ha realizado él mismo. Los demás sólo pueden borrar las reservas realizadas por ellos mismos.



- b. Usuarios. Sólo pueden realizar las bajas de usuario los usuarios *administradores*.
 - c. Noticias y agenda.
2. Desconectarse, salir de la sesión de usuario registrado. Lo puede realizar un usuario *registrado*.
 3. Que la aplicación soporte tres idiomas, catalán, inglés y español. Es decir, se podrá elegir el idioma en el que se desea ver la aplicación.

Para solucionar el problema, se ha diseñado una aplicación Web que utiliza una arquitectura basada en el patrón MVC y la implementación se realiza empleando Struts e Hibernate, las cuales se explica detalladamente en el *capítulo 2, Arquitectura del Sistema*.

A continuación, mostramos un análisis detallado de los requisitos, lo cuales fueron extraídos a partir del documento de propuesta, reuniones con el cliente y directora del proyecto.



4. SOLUCIÓN

En este apartado se comenta la solución que se ha desarrollado para solucionar la problemática que nos planteaba el cliente, este capítulo consta de los siguientes apartados:

- Análisis e Identificación de los requisitos.
- Diseño de la aplicación
- Implementación de la aplicación.

4.1 ANÁLISIS DE LOS REQUISITOS

El propósito de la definición de requisitos es analizar la declaración de los requisitos de usuario y producir un conjunto de requisitos software tan completo, constante y correcto como sea posible.

Los requisitos definen “qué” debe hacer el producto. Son una referencia para verificar el diseño y el producto. Los aspectos relativos al “cómo” no deben incluirse excepto aquellos que restringen las capacidades del software.

Para esto se analiza el problema construyendo una descripción coherente y completa de lo que el software deberá hacer. Los requisitos de software son una visión de problema vista por el desarrollador, antes que el usuario.

Se va a construir un modelo de implementación independiente de lo que necesita el usuario. Este se llama un modelo lógico, y se utiliza para producir los requisitos de software.

El modelo lógico puede ser construido por descomposición de arriba debajo de la función principal, o inferido de los requisitos de usuario mediante una jerarquía de funciones. El modelado es un proceso iterativo. Las partes del modelo pueden necesitar volver a ser especificadas muchas veces antes de lograr una descripción completa, coherente y constante.

Los requisitos obtenidos se van a dividir en dos, dependiendo del grado de implicación que éstos tienen en la funcionalidad de la aplicación a ser desarrollada. Estos dos tipos de requisitos son:



- Requisitos Funcionales. Estos especifican “qué” tiene que hacer el software. Definen el propósito del software. Los requisitos funcionales se derivan del modelo lógico, que a su vez se deriva de los **requisitos de capacidad de usuario**. Para que puedan ser indicados cuantitativamente, los requisitos funcionales deben incluir atributos del desempeño.
- Requisitos de Interfaz. Estos especifican el hardware, los elementos de software o base de datos con que el sistema, o componente del mismo, deben de actuar y/o comunicarse. Los requisitos de Interfaz se clasifican en
 - Software
 - Hardware
 - Comunicaciones
- Requisitos de Seguridad. Especifican los requisitos para asegurar el sistema contra amenazas de confidencialidad, integridad y disponibilidad, por ejemplo inhibidores de órdenes, acceso de sólo lectura, sistema de contraseña, protección de virus, etc.
- Requisitos Operacionales. Estos especifican cómo el sistema correrá y cómo comunicará con los usuarios. Los requisitos operacionales incluyen al interfaz hombre-maquina así como los requisitos logísticos y organizacionales. Como requisitos operacionales se pueden indicar los siguientes ejemplos: la disposición de pantalla, el contenido de mensajes de error, de sistemas de ayuda, etc.

4.2 IDENTIFICACIÓN DE REQUISITOS

La tarea de identificación de requisitos se lleva a cabo mediante reuniones, e-mails, distendidas con un responsable de la universidad de Lleida y por la tutora del proyecto.

Para que la recogida de requisitos se realice de forma clara, sencilla y estructurada se ha definido una plantilla con las siguientes propiedades:



IDENTIFICADOR:	
Nombre:	
Descripción:	
Prioridad: <input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	
Fuente:	

Donde:

- **Identificador:** Identifica de forma unívoca cada uno de los requisitos. Dicho identificador sigue la siguiente nomenclatura.

RU- Tipo- Número

Donde:

- Tipo: puede tomar los valores:
 - C si se trata de un requisito de usuario de Capacidad o Funcionalidad.
 - R si se trata de un requisito de usuario de Restricción.
 - I si es de Interfaz.
 - S si es de Seguridad.
 - P si es de Portabilidad.
- Número: será un número de tres cifras que empezará por 001 y se irá incrementado en una unidad por cada nuevo requisito añadido.

Ej. *RUR010*: Requisito de Usuario de Restricción, Número 10

RUC010: Requisito de Usuario de Capacidad, Número 10

- **Nombre del requisito:** Expresa en pocas palabras un resumen del requisito.
- **Descripción:** Breve comentario textual del requisito.



- **Prioridad:** Indica la prioridad en el desarrollo del requisito. Puede tomar los valores “Alta”, “Media” o “Baja”.
- **Necesidad:** Indica el nivel de necesidad del requisito dentro del sistema final. Puede tomar los valores “Esencial”, “Opcional” o “Conveniente”. Un requisito será “Esencial”, cuando el cliente no acepte ninguna negociación, “Conveniente”, cuando el requisito se pueda negociar u “Opcional” cuando su implementación no es imprescindible.
- **Estabilidad:** Indica la posibilidad de que el requisito cambie a lo largo del desarrollo de la aplicación. Puede tomar los valores “Si”, cuando el cliente asegura que no va a ser modificado o “No”, cuando el requisito puede variar en función de las sucesivas etapas del proyecto.
- **Prerequisito:** Indica si necesita algún requisito para poderse realizar.
- **Fuente:** Indica el origen a partir del cual se ha obtenido el documento. Cuando se trata de un documento externo se hace referencia a dicho documento.



4.2.1 REQUISITOS DE CAPACIDAD

IDENTIFICADOR: RUC001	
Nombre:	Consulta la parte pública del portal.
Descripción:	Los usuarios no registrados pueden consultar todas las opciones mostradas en el portal, sin necesidad de identificarse en la aplicación.
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	Ninguno
Fuente:	Documento de propuesta

IDENTIFICADOR: RUC002	
Nombre:	Acceso a zona restringida por usuarios registrados en la aplicación
Descripción:	Los usuarios registrados pueden consultar/añadir información en la zona restringida del portal, para ello el sistema deberá validar su usuario y password.
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	Estar dado de alta en la aplicación y ser validado en la aplicación principal
Fuente:	Documento de propuesta.



IDENTIFICADOR: RUC003	
Nombre:	Consultar información del departamento: noticias y agenda.
Descripción:	<p>Los usuarios sin necesidad de registrarse pueden consultar información del departamento:</p> <ul style="list-style-type: none">▪ Se encuentra en la <i>sección noticias y agenda</i> de la pantalla principal. <p>Esta información se mostrará en la página principal de usuarios registrados en sus correspondientes secciones.</p> <p>La información de la agenda se refiere a eventos que se van a producir en el departamento. Aparece:</p> <ul style="list-style-type: none">• Día: Fecha del evento.• Título: Nombre del evento.• Hora: en la que se producirá el evento.• Enlace a más información: página Web externa a la aplicación que contiene información adicional del evento en cuestión. <p>La información de las noticias se refiere a la información de interés que desean publicar los miembros del departamento. Aparece:</p> <ul style="list-style-type: none">• Día: Fecha de la noticia.• Título de la noticia.• Descripción: una breve descripción de la misma.• Enlace a más información: página Web externa a la aplicación que contiene información adicional de la noticia en cuestión. <p>Como máximo se mostrarán 4 noticias e información de agenda en cada sección. Corresponderán a la información del mes en curso.</p>
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	Ninguno.
Fuente:	Documento de propuesta y reuniones con el cliente.



IDENTIFICADOR: RUC004	
Nombre:	Consultar información del departamento: “¿Quiénes somos?”
Descripción:	<p>Los usuarios sin necesidad de registrarse pueden consultar información del departamento y acceder a un video del mismo.</p> <p>La información que se muestra corresponde al grupo del departamento. Es decir, es una presentación del mismo.</p>
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	Ninguno
Fuente:	Documento de propuesta

IDENTIFICADOR: RUC005	
Nombre:	Consultar información del departamento: De todo el personal
Descripción:	<p>Los usuarios sin necesidad de registrarse pueden consultar información del personal del departamento y acceder a sus currículos.</p> <p>Esta información de los usuarios se corresponde a la añadida en la Base de Datos del departamento, como usuarios del sistema.</p> <p>La información asociada a sus currículos nos las proporcionará el departamento de la universidad, ya que la tienen creada de manera independiente al sistema desarrollado.</p>
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	Ninguno
Fuente:	Documento de propuesta



IDENTIFICADOR: RUC006	
Nombre:	Consultar información del departamento: Personal PDI
Descripción:	<p>Los usuarios sin necesidad de registrarse pueden consultar información del personal PDI del departamento y acceder a sus currículos.</p> <p>Esta información de los usuarios se corresponde a la añadida en la Base de Datos del departamento, como usuarios del sistema que pertenecen al grupo de personal PDI.</p> <p>La información asociada al Personal PDI, son los catedráticos, profesores titulares y ayudantes del departamento.</p>
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	Ninguno
Fuente:	Documento de propuesta

IDENTIFICADOR: RUC007	
Nombre:	Consultar información del departamento: Personal PAS (Personal Administrativo)
Descripción:	<p>Los usuarios sin necesidad de registrarse pueden consultar información del personal PAS del departamento y acceder a sus currículos.</p> <p>Esta información de los usuarios se corresponde a la añadida en la Base de Datos del departamento, como usuarios del sistema que pertenecen al grupo de personal PAS.</p>
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	Ninguno
Fuente:	Documento de propuesta



IDENTIFICADOR: RUC008	
Nombre:	Consultar información del departamento: Personal Becarios
Descripción:	<p>Los usuarios sin necesidad de registrarse pueden consultar información del personal Becarios del departamento y acceder a sus currículos.</p> <p>Esta información de los usuarios se corresponde a la añadida en la Base de Datos del departamento, como usuarios del sistema que pertenecen al grupo de personal Becarios del departamento.</p>
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	Ninguno
Fuente:	Documento de propuesta

IDENTIFICADOR: RUC009	
Nombre:	Consultar información de la docencia que se imparte en el departamento.
Descripción:	<p>Los usuarios sin necesidad de registrarse pueden consultar información de la docencia que imparte el departamento:</p> <ol style="list-style-type: none">1. Titulaciones:<ul style="list-style-type: none">• Nombre de la titulación.• Conjunto de las asignaturas que tiene, con posible enlace de acceso a más información de las mismas. Se muestra con un archivo en formato “pdf” o mediante una “url” asociada.2. Masters:<ul style="list-style-type: none">• Nombre del master.• Acceso a más información del mismo mediante una “url” asociada.3. Cursos de especialización:<ul style="list-style-type: none">• Nombre del curso.• Acceso a más información del mismo mediante una “url” asociada.
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	



IDENTIFICADOR: RUC009	
Prerequisito:	Ninguno
Fuente:	Documento de propuesta

IDENTIFICADOR: RUC010	
Nombre:	Consultar información de los Proyectos Finales de Carrera (PFC), Becas y trabajos dirigidos que se imparte en el departamento.
Descripción:	<p>Los usuarios sin necesidad de registrarse pueden consultar información de las ofertas del departamento:</p> <ol style="list-style-type: none">Proyecto Final de Carrera (PFC):<ul style="list-style-type: none">• Título del proyecto.• Descripción.• Coordinador que lo tutoriza.• E-mail de contacto del tutor.Becas:<ul style="list-style-type: none">• Nombre de la beca.• Descripción de la misma.• Url asociada a la beca para poder mostrar más información.• Licenciatura exigida para poder solicitar la beca.• Cuantía de beca.Trabajos Dirigidos:<ul style="list-style-type: none">• Título del trabajo.• Descripción.• Coordinador que lo tutoriza.• E-mail de contacto del tutor.
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	Ninguno
Fuente:	Documento de propuesta y reuniones con el cliente



IDENTIFICADOR: RUC011	
Nombre:	Consultar información de las tareas de investigación que desempeña el departamento.
Descripción:	<p>Los usuarios sin necesidad de registrarse pueden consultar información de las tareas de investigación del departamento:</p> <ol style="list-style-type: none">Grupos de investigación que desarrolla el departamento:<ul style="list-style-type: none">Tipo de grupo de investigación:Por cada tipo grupo de investigación:<ul style="list-style-type: none">Nombre del grupo de investigación.Acceso a información adicional a través de un enlace asociado.Línea de investigación del grupo.Tesis doctorales realizadas en el departamento:<ul style="list-style-type: none">Título de la tesis.Autor.Descripción de la tesis.Universidad en la que se realiza la tesis.Año de la defensa.Más información asociada. Se añade un enlace a una página Web con la información.Transferencia tecnológica: Servicios que contrata la universidad con otras empresas. Se pide que se muestre la información de<ul style="list-style-type: none">Nombre del servicio que se contrata.Empresa que lo realiza.Periodo que dura el servicio contratado.
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	Ninguno
Fuente:	Documento de propuesta y reuniones con el cliente



IDENTIFICADOR: RUC012	
Nombre:	Cambiar la contraseña de un usuario.
Descripción:	Los usuarios registrados pueden cambiar su contraseña. Para ello se necesitará añadir el identificador de usuario dentro del sistema.
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	Estar dado de alta en el portal Web como un usuario registrado.
Fuente:	Documento de propuesta.

IDENTIFICADOR: RUC013	
Nombre:	Gestionar documentos.
Descripción:	<p>Los usuarios registrados pueden consultar documentos de actas o consejos y subirlos.</p> <p>Se mostrará el tipo de documento que es y haciendo “click” sobre el propio nombre se podrá acceder a la información del documento.</p> <p>Este documento puede estar en una página Web o bien puede ser un documento de texto, “pdf”...que el sistema deberá de copiar y dejarlo almacenado en el portal Web.</p>
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	Estar dado de alta en la aplicación
Fuente:	Propuesta de la tutora del proyecto.



IDENTIFICADOR: RUC014	
Nombre:	Gestionar reservas.
Descripción:	Los usuarios registrados pueden consultar, modificar, borrar y realizar reservas. Los tipos de reservas son: <ol style="list-style-type: none">1. Reserva de ordenador.2. Reserva de laboratorio.3. Reserva de cañón.4. Reserva de coche.
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	Estar dado de alta en la aplicación
Fuente:	Documento de propuesta y reuniones con el cliente.

IDENTIFICADOR: RUC015	
Nombre:	Añadir reservas de ordenador, laboratorio, cañón y coche.
Descripción:	Los usuarios registrados pueden realizar reservas, los datos a rellenar son: <ol style="list-style-type: none">1. Identificar el objeto a reservar.2. Fecha de la reserva.3. Duración de la reserva. Se debe controlar que un mismo objeto no puede ser reservado para una misma fecha y hora. En el caso de ser así, se le informará al usuario mediante un mensaje de aviso.
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	Estar dado de alta en la aplicación
Fuente:	Documento de propuesta y reuniones con el cliente.



IDENTIFICADOR: RUC016	
Nombre:	Modificar reservas de ordenador, laboratorio, cañón y coche.
Descripción:	<p>Los usuarios registrados pueden modificar las reservas que han realizado.</p> <p>Si es un usuario administrador podrá modificar cualquier reserva de cualquier usuario.</p> <p>El único campo que se permite modificar es la <i>fecha</i>. Si para esa fecha, hora y objeto a reservar, existe una reserva, se le informará al usuario. Y también se informará al usuario la imposibilidad de modificar una reserva que no ha realizado él mismo.</p>
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	Estar dado de alta en la aplicación y existir la reserva correspondiente
Fuente:	Documento de propuesta y reuniones con el cliente.

IDENTIFICADOR: RUC017	
Nombre:	Borrar reservas de ordenador, laboratorio, cañón y coche.
Descripción:	<p>Los usuarios registrados pueden borrar las reservas que han realizado.</p> <p>Si es un usuario administrador podrá borrar cualquier reserva de cualquier usuario.</p> <p>Si se ha borrado correctamente, se informará al usuario del éxito de la operación. Al igual que en las modificaciones se informará al usuario la imposibilidad de querer borrar una reserva que no ha realizado él mismo.</p>
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	Estar dado de alta en la aplicación y existir la reserva correspondiente.
Fuente:	Documento de propuesta y reuniones con el cliente.



IDENTIFICADOR: RUC018	
Nombre:	Alta de los usuarios de la aplicación.
Descripción:	<p>El administrador del sistema puede dar de alta a los usuarios de la aplicación, es decir, puede añadir miembros del departamento nuevos. Los datos que se insertan en un alta son:</p> <ul style="list-style-type: none">• Nombre del usuario.• Apellidos del usuario.• Tipo de usuario: PAS (Personal Administrativo), Becario, PDI-Catedrático, PDI-Ayudante, PDI-Titular.• DNI: NIF del usuario.• Contraseña de acceso al sistema.• URL: dirección Web en la que se encuentra la página con el currículo asociado. <p>No se puede dar de alta a dos usuarios con el mismo D.N.I..</p>
Prioridad: <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	El administrador del sistema debe estar dado de alta en la aplicación
Fuente:	Documento de propuesta.



IDENTIFICADOR: RUC019	
Nombre:	Modificar información de los usuarios de la aplicación.
Descripción:	<p>Todos los usuarios podrán modificar sus datos personales. Que son:</p> <ul style="list-style-type: none">• Nombre del usuario.• Apellidos del usuario.• Tipo de usuario: PAS (Personal Administrativo), Becario, PDI-Catedrático, PDI-Ayudante, PDI-Titular.• DNI: NIF del usuario.• Contraseña de acceso al sistema.• URL: dirección Web en la que se encuentra la página con el currículum asociado. <p>El administrador del sistema puede modificar los datos personales de cualquier usuario de la aplicación.</p>
Prioridad: <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	El usuario debe estar dado de alta en la aplicación
Fuente:	Documento de propuesta y reuniones con el cliente.

IDENTIFICADOR: RUC020	
Nombre:	Borrar usuarios de la aplicación.
Descripción:	<p>El administrador del sistema puede borrar cualquier usuario de la aplicación excepto él mismo.</p> <p>La consecuencia es una baja en la aplicación por lo que no podrá acceder como usuario registrado. Pero no se podrá dar la baja usuarios que contengan reservas en la aplicación. Para poder borrarlo, previamente se ha de borrar todas las reservas que haya realizado.</p>
Prioridad: <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	El usuario debe estar dado de alta en la aplicación
Fuente:	Reuniones con el cliente.



IDENTIFICADOR: RUC021	
Nombre:	Consultar información del departamento: noticias y agenda.
Descripción:	<p>Los usuarios registrados pueden consultar información interna del departamento: noticias y agenda. Esta información se corresponde con las noticias e información de la agenda del departamento y que es sólo para los miembros que lo componen.</p> <p>Esta información se mostrará en la página principal de usuarios registrados en sus correspondientes secciones.</p> <p>Si es información de la agenda, aparece:</p> <ul style="list-style-type: none">• Día: Fecha de la noticia.• Título: Nombre del evento.• Hora: hora en la que se producirá el evento.• Enlace a más información: página Web externa a la aplicación que contiene información adicional del tema en cuestión. <p>Y si es una noticia, aparece:</p> <ul style="list-style-type: none">• Día: Fecha de la noticia.• Título de la noticia• Descripción: una breve descripción de la misma.• Enlace a más información: página Web externa a la aplicación que contiene información adicional del tema en cuestión. <p>Como máximo se mostrarán 4 noticias e información de agenda en cada sección. Corresponderán a la información del mes en curso.</p>
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	El usuario debe estar registrado y situado en la página principal de los usuarios registrados.
Fuente:	Documento de propuesta y reuniones con el cliente.



IDENTIFICADOR: RUC022	
Nombre:	Añadir información: noticias y agenda.
Descripción:	Los usuarios registrados pueden añadir información interna y pública. Si añaden información interna, se mostrará en la página principal de los usuarios registrados y si la información añadida es pública, se mostrará en la página principal de la aplicación. Esta información se puede añadir en la sección de agenda o de noticias.
Prioridad: <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	El usuario debe estar validado.
Fuente:	Reuniones con el cliente.

IDENTIFICADOR: RUC023	
Nombre:	Añadir información de agenda.
Descripción:	Los usuarios registrados pueden añadir información en la sección de agenda tanto interna como pública. Si añaden información interna, se mostrará en la página principal de los usuarios registrados y si la información añadida es publica, se mostrará en la página principal de la aplicación. La información que se debe insertar para añadirla correctamente es: <ul style="list-style-type: none">• Título del evento.• Si la información se muestra en la sección interna o externa.• URL: se asocia a las noticias una página Web en la que se detalla la información. Esta página es independiente de la aplicación desarrollada.• Hora: Indica la hora en la que se producirá el evento.• Fecha: Fecha en la que se produce le evento. Para facilitar la inserción de esta información, debe aparecer una opción de ayuda donde se describe cada uno de los campos a rellenar. Esta información ha de estar triplicada en tres idiomas: Inglés, castellano y catalán.
Prioridad: <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	El usuario debe estar validado.
Fuente:	Documento de propuesta y reuniones con el cliente.



IDENTIFICADOR: RUC024	
Nombre:	Añadir información de noticias.
Descripción:	<p>Los usuarios registrados pueden añadir información en la sección de agenda tanto interna como pública.</p> <p>Si añaden información interna, se mostrará en la página principal de los usuarios registrados y si la información añadida es publica, se mostrará en la página principal de la aplicación.</p> <p>La información que se debe insertar para añadirla correctamente es:</p> <ul style="list-style-type: none">• Título de la noticia.• Si la información se muestra en la sección interna o externa.• URL: se asocia a las noticias una página Web en la que se detalla la información. Esta página es independiente de la aplicación desarrollada.• Descripción: breve descripción de la noticia en cuestión.• Fecha: Fecha en la que se produce la noticia. <p>Para facilitar la inserción de esta información, debe aparecer una opción de ayuda donde se describe cada uno de los campos a rellenar.</p> <p>Esta información ha de estar triplicada en tres idiomas: Inglés, castellano y catalán.</p>
Prioridad: <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	El usuario debe estar validado.
Fuente:	Documento de propuesta y reuniones con el cliente.



IDENTIFICADOR: RUC025	
Nombre:	Borrar información de noticias y agenda.
Descripción:	Los usuarios registrados pueden borrar información en la sección de agenda y noticias tanto interna como pública. Esta información se borrará en tres idiomas: Inglés, castellano y catalán. Con la finalidad de no provocar inconsistencias en la Base de Datos.
Prioridad: <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	El usuario debe estar validado.
Fuente:	Documento de propuesta y reuniones con el cliente.

IDENTIFICADOR: RUC026	
Nombre:	La aplicación soporta 3 idiomas: inglés, español y catalán.
Descripción:	Al inicio de la aplicación, en la página principal el usuario puede elegir el idioma para visualizar la información del portal Web.
Prioridad: <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	Ninguno.
Fuente:	Documento de propuesta y reuniones con el cliente.

IDENTIFICADOR: RUC027	
Nombre:	Modificar información de noticias y agenda.
Descripción:	Los usuarios registrados pueden modificar información en la sección de agenda y noticias tanto interna como pública. Esta información se modificará en tres idiomas. Las modificaciones las pueden realizar cualquiera de los usuarios registrados.
Prioridad: <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	El usuario debe estar validado.
Fuente:	Documento de propuesta y reuniones con el cliente.



IDENTIFICADOR: RUC028	
Nombre:	La aplicación debe mostrar opción de ayudas.
Descripción:	Los usuarios registrados cuando quieren realizar operaciones de inserciones, modificaciones o borrados, deben de tener una opción de ayuda para que se les informe detalladamente de los campos a rellenar, restricciones, etc.
Prioridad: <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	El usuario debe estar validado.
Fuente:	Autora del proyecto.

IDENTIFICADOR: RUC029	
Nombre:	La aplicación debe mostrar mensajes de error.
Descripción:	Los usuarios registrados cuando quieren realizar operaciones y no se realizan con éxito, deben de estar informados mediante mensajes de error.
Prioridad: <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	El usuario debe estar validado.
Fuente:	Autora del proyecto.



4.2.2 REQUISITOS DE SEGURIDAD

IDENTIFICADOR: RUS001	
Nombre:	Sistema protegido por usuario y contraseña para acceder a la parte privada.
Descripción:	Identificación del usuario: <ul style="list-style-type: none">▪ Usuario: corresponde con el DNI del usuario y permite caracteres alfanuméricos con máximo de 15.▪ Clave: caracteres alfanuméricos con máximo de 15.
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	Ninguno.
Fuente:	Autora del proyecto



4.2.3 REQUISITOS DE INTERFAZ

IDENTIFICADOR: RUI001	
Nombre:	Formato de todas las páginas de la aplicación.
Descripción:	<p>Las páginas de la aplicación se dividirá en 3 zonas: zona superior, margen izquierdo y zona central.</p> <div><div><div>ZONA SUPERIOR</div><div>Universidad de Lleida Escuela Técnica Superior de Ingeniería Agraria Departamento de Producción Vegetal y Ciencia Forestal</div></div><div><div>M A R G E N I Z Q U I E R D O</div><div>ZONA CENTRAL</div></div></div>
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	Ninguno.
Fuente:	Documento de propuesta



IDENTIFICADOR: RUI002	
Nombre:	Formato de las páginas principal: Zona superior.
Descripción:	<p>La zona superior será fija en toda las páginas y mostrará el logo con el nombre de la Universidad, de la Escuela y del departamento. El tamaño de la letra irá incrementando con un aspecto similar a:</p> <p style="text-align: center;">Universidad de Lleida Escuela Técnica Superior de Ingeniería Agraria Departamento de Producción Vegetal y Ciencia Forestal</p> <p>En los laterales a estas letras, se añadirán unas imágenes asociadas al logo de la universidad y al departamento de investigación que incluyen enlaces a los correspondientes sitios Web.</p>
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	Ninguno.
Fuente:	Documento de propuesta

IDENTIFICADOR: RUI003	
Nombre:	Formato de las páginas: Zona central.
Descripción:	<p>En la zona central aparecerá información del departamento.</p> <p>Inicialmente aparecerá una agenda y un conjunto de noticias. Cuando se seleccione algún <i>link</i> del margen izquierdo, aparecerá información referente a la petición solicitada. Empezando con un <i>Título principal</i> que alude a la opción seleccionada.</p>
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	Ninguno.
Fuente:	Documento de propuesta



IDENTIFICADOR: RUI004	
Nombre:	Distinto formato del Margen izquierdo de las páginas que utilizan los <u>usuarios no registrados</u> frente a los <u>registrados</u>.
Descripción:	<p>El menú que muestra la página de usuarios registrados es distinto al menú que muestran los usuarios no registrados.</p> <p>Casi todas o todas las funcionalidades que puede realizar un usuario en el portal Web se encuentran en el margen izquierdo. Además dependiendo de si un usuario es registrado o no podrá realizar más o menos funcionalidades del portal, implica que las opciones de menú deben ser distintas para un usuario registrado frente a los no registrados. Así que el menú de las paginas principales de usuario registrado y no registrado son distintas.</p>
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	Ninguno.
Fuente:	Documento de propuesta



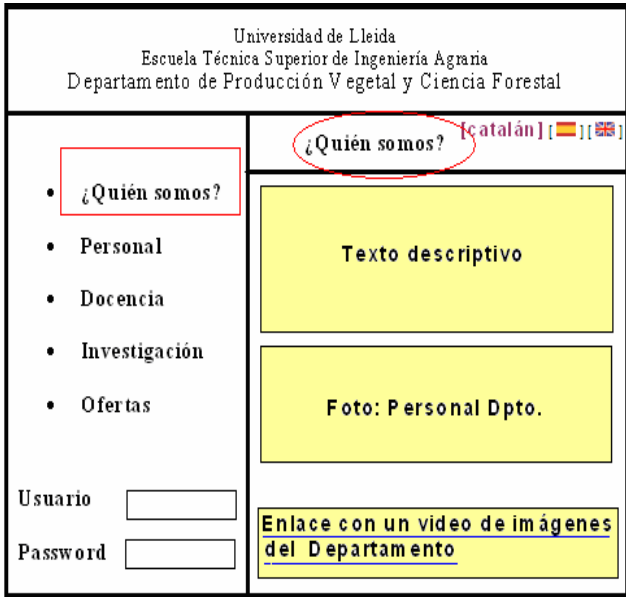
IDENTIFICADOR: RUI004	
Nombre:	Formato de las páginas que utilizan los <u>usuarios no registrados</u>: Margen izquierdo.
Descripción:	<p>En el margen izquierdo se mostrará un menú fijo con las operaciones a realizar que son:</p> <ul style="list-style-type: none">• <i>¿Quién somos?</i>• <i>Personal</i>: desplegable con : PDI (Personal docente investigador), PAS (Personal de administración y servicios) y Becarios• <i>Docencia</i>: desplegable con: Docencia de Grado, Masteres, Cursos especialización• <i>Investigación</i>: Grupos de investigación, Tesis doctorales, transferencia de tecnología• <i>Ofertas</i>: desplegable con: Becas, Proyectos Final de Carrera, Trabajos• <i>Acceso restringido</i> mediante: Nombre de <i>usuario</i> y <i>password</i>.• <i>Opción de cambiar la contraseña</i>.
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	Ninguno.
Fuente:	Documento de propuesta



IDENTIFICADOR: RUI005	
Nombre:	Formato de las páginas que utilizan los <u>usuarios registrados</u>: Margen izquierdo.
Descripción:	<p>En el margen izquierdo se mostrará un menú fijo con las operaciones a realizar por los usuarios registrados que son:</p> <ul style="list-style-type: none">• <i>Documentos</i>: desplegable para visualizar documentos de: actas o consejos y opción de subir documentos.• <i>Reservas</i>: desplegable con las operaciones de: ver reservas, hacer reservas, modificar y borrar.• <i>Usuario</i>: para los usuarios “no administradores” se mostrará la opción de modificar sus datos. Si es un usuario administrador, además de la opción de modificar, se muestra las opciones de dar el alta y baja de usuarios.• <i>Información en noticias</i>: Se puede añadir, borrar y modificar información pública o privada.• <i>Información en agenda</i>: Se puede añadir, borrar y modificar información pública o privada.• <i>Volver a la página de acceso público</i>: vuelve a la página principal de usuarios registrados.
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input type="checkbox"/> Si <input checked="" type="checkbox"/> No	
Prerequisito:	Ninguno.
Fuente:	Documento de propuesta y reuniones con el cliente.





IDENTIFICADOR: RUI006	
Nombre:	Formato de las páginas de <u>acceso restringido</u>.
Descripción:	<p>En la todas las páginas se debe dar la <i>bienvenida</i> al usuario de la sesión en curso y la posibilidad de <i>salir</i> de la sesión en todo momento.</p> <ul style="list-style-type: none"> En el margen izquierdo se mostrará lo comentado en el requisito anterior, RUI005. La cabecera de las páginas será la misma que la de los usuarios no registrados. La zona central mantendrá el mismo estilo que las páginas de los usuarios no registrados.
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input type="checkbox"/> Si <input checked="" type="checkbox"/> No	
Prerequisito:	Estar registrado en la aplicación.
Fuente:	Documento de propuesta y reuniones con el cliente.

IDENTIFICADOR: RUI007	
Nombre:	Formato de la página que muestra la opción de menú: ¿Quiénes somos?
Descripción:	<p>Se desea que muestre:</p> 
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente



IDENTIFICADOR: RUI007	
Estabilidad: <input type="checkbox"/> Si <input checked="" type="checkbox"/> No	
Prerequisito:	Ninguno.
Fuente:	Documento de propuesta

IDENTIFICADOR: RUI008	
Nombre:	Formato de la página que muestra la opción de menú: Personal
Descripción:	<p>Se desea que muestre:</p> <div><div>Universidad de Lleida Escuela Técnica Superior de Ingeniería Agraria Departamento de Producción Vegetal y Ciencia Forestal</div><div><div><ul style="list-style-type: none">¿Quién somos?PersonalDocenciaInvestigaciónOfertas<div>Usuario <input type="text"/></div><div>Password <input type="text"/></div></div><div><div>Personal [catalán] [] []</div><div>Texto descriptivo</div><div>Datos Miembro..... Cargo Datos Miembro..... Cargo Datos Miembro..... Cargo Datos Miembro..... Cargo Datos Miembro..... Cargo Datos Miembro..... Cargo Datos Miembro..... Cargo Datos Miembro..... Cargo</div></div></div></div>
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input type="checkbox"/> Si <input checked="" type="checkbox"/> No	
Prerequisito:	Ninguno.
Fuente:	Documento de propuesta



IDENTIFICADOR: RUI009	
Nombre:	Formato de información que se muestra al seleccionar la opción de Investigación
Descripción:	El resultado de la operación muestra una salida similar a esta, compuesta por: <ul style="list-style-type: none">▪ Texto descriptivo.▪ Las tres posibles líneas de investigación, las cuales tienen un enlace a un sitio Web.
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input type="checkbox"/> Si <input checked="" type="checkbox"/> No	
Prerequisito:	Ninguno.
Fuente:	Documento de propuesta



4.2.4 REQUISITOS DE PORTABILIDAD

IDENTIFICADOR: RUP001	
Nombre:	Navegadores de la aplicación
Descripción:	Esta aplicación funciona correctamente para el navegador Internet Explorer. versiones
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	Ninguno.
Fuente:	Documento de propuesta

IDENTIFICADOR: RUP002	
Nombre:	Portabilidad de la aplicación
Descripción:	Esta aplicación se puede implantar en los sistemas operativos Windows (95, 98, 2000, XP) y Linux gracias a la utilización de JAVA como lenguaje de programación.
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	Ninguno.
Fuente:	Documento de propuesta

IDENTIFICADOR: RUP002	
Nombre:	Requisitos de funcionamiento
Descripción:	Esta aplicación requiere: <ul style="list-style-type: none">○ Base de Datos MySql 5.2.3.○ Servidor Apache, Tomcat 5.5.
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
Estabilidad: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Prerequisito:	Ninguno.
Fuente:	Documento de propuesta.

4.2.5 DIAGRAMA DE CASOS DE USO

Del análisis de requisitos, obtenemos los siguientes casos de uso con los que mostramos las funcionalidades que pueden realizar los usuarios, en función de si se han registrado o no en el portal Web. Comenzamos con los usuarios no registrados, definidos como usuarios públicos:

- **Usuario Público:** incluye las operaciones de visualización correspondientes a las consultas de información.

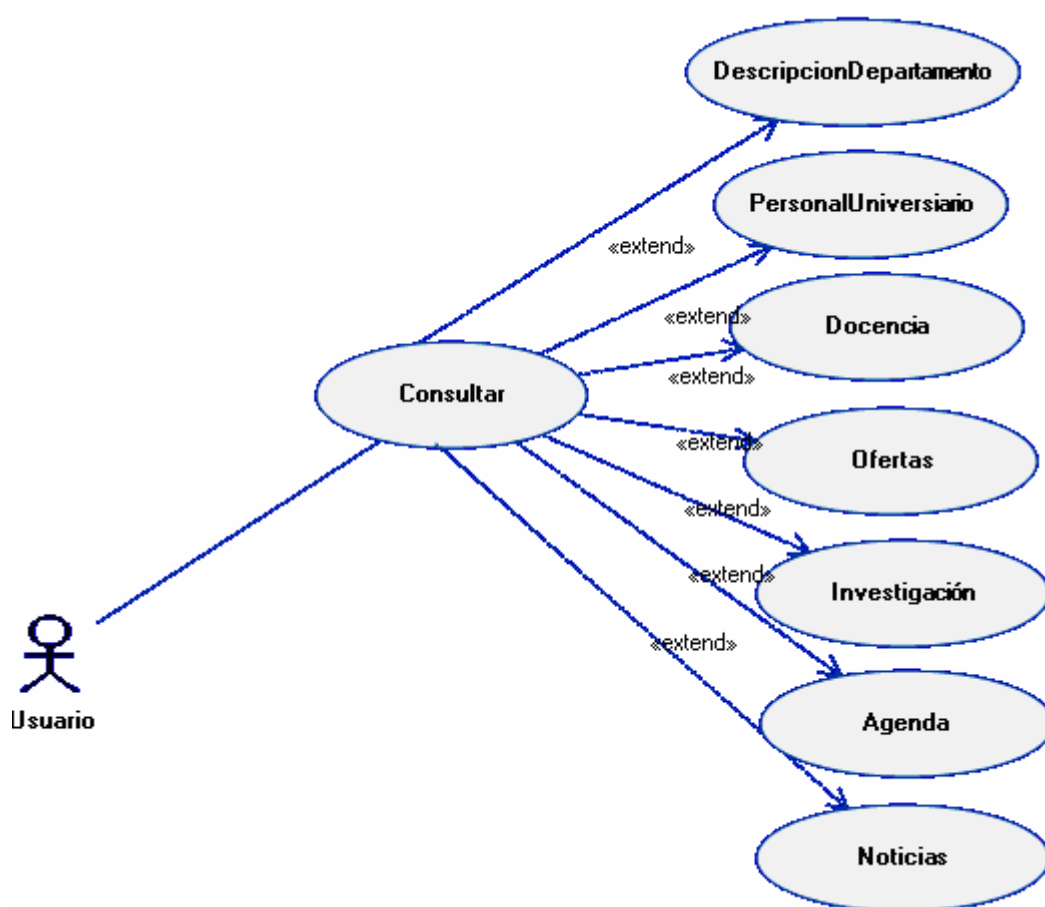


Figura 4-1: Casos de uso-Usuario Público

- Un **usuario registrado** también puede realizar las operaciones de un usuario público, así que incluimos todas las que puede realizar. Los casos de uso que están en rojo corresponden a las funciones exclusivas de los usuarios registrados.

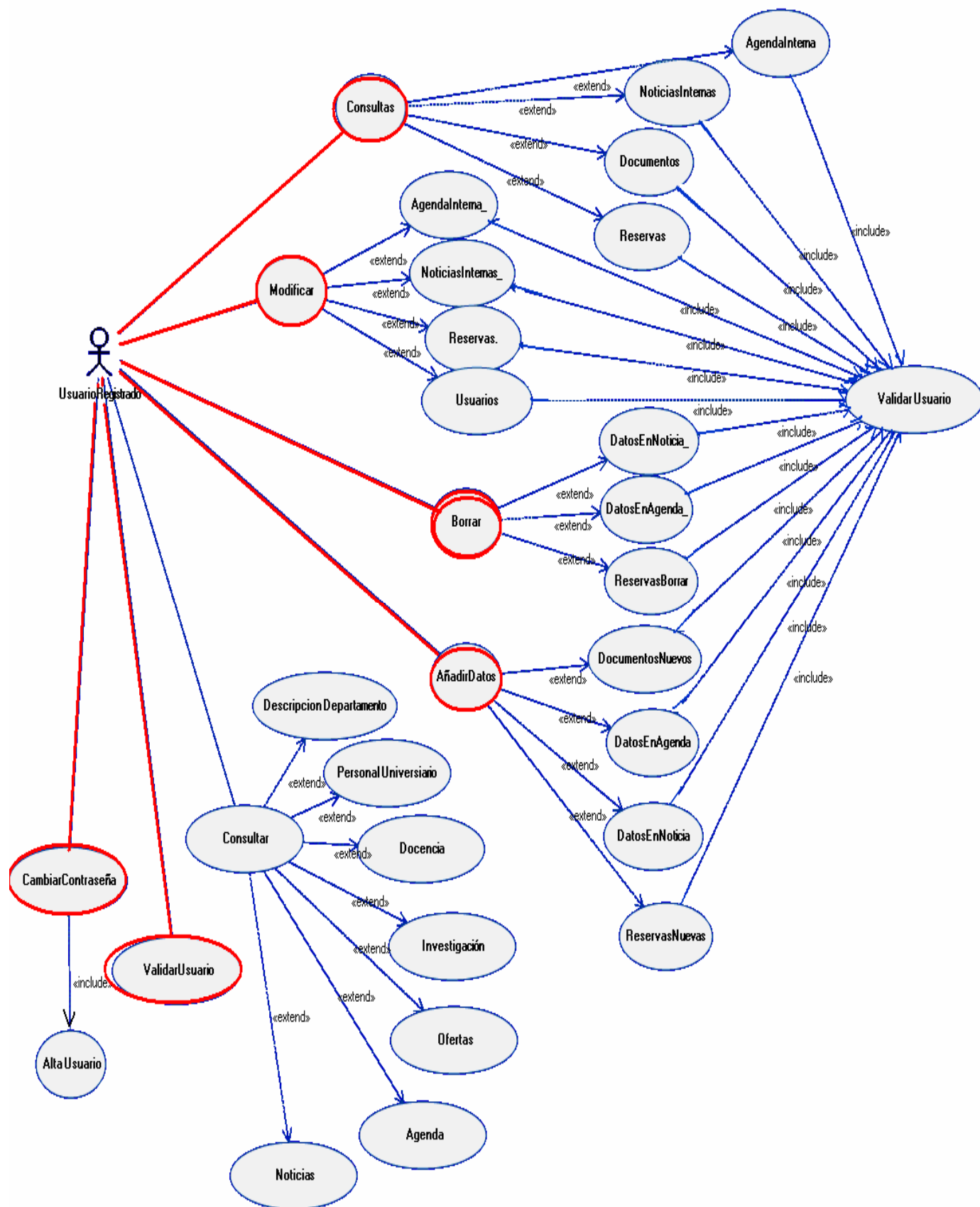


Figura 4-2: Casos de uso- Usuario Registrado

- Un usuario registrado con el rol de administrador, puede realizar dos funcionalidades más: altas de usuarios y bajas.

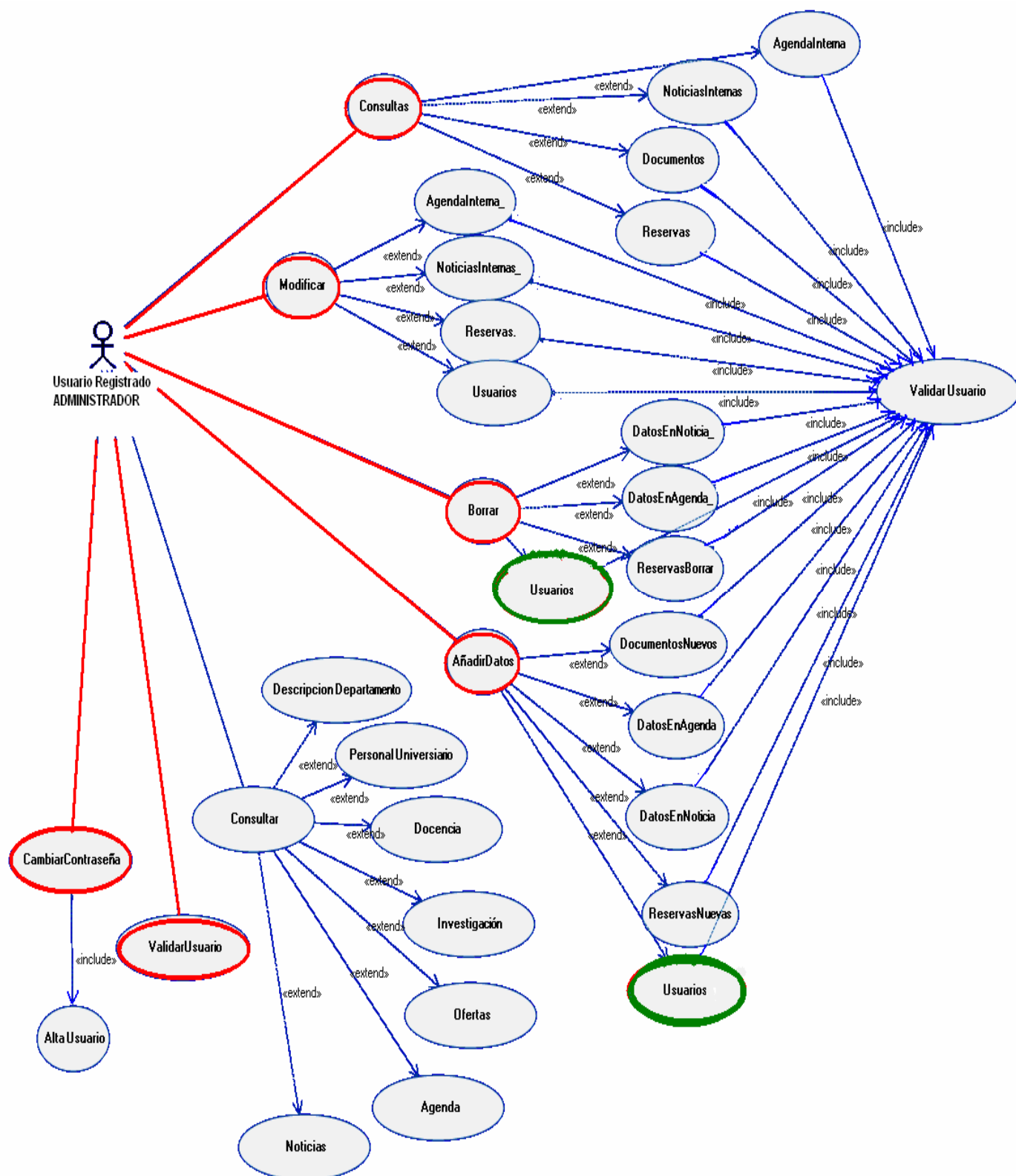


Figura 4-3: Casos de uso-Usuario Administrador



Los casos de uso subrayados en verde, son los que puede realizar un **administrador**:

- Puede dar **altas** de usuarios
- Dar **bajas**, excepto la suya propia.
- Y respecto a las **reservas**, puede modificar cualquier reserva y borrarla, independientemente de si la ha realizado el administrador o no.



4.3 DISEÑO

Una vez comentados los requisitos y contruidos los casos de uso en base a las especificaciones de requisitos, se aclaran las funcionalidades a realizar dependiendo del tipo de usuario que esté en la aplicación. Por lo tanto, comenzamos con la parte del diseño de la solución, para ello se elige una arquitectura, se define la estructuración de las páginas Web en base a la distribución de información y se diseña la Base de Datos que almacena la información.

4.3.1 ARQUITECTURA MVC

Como hemos mencionado anteriormente, la arquitectura que se ha utilizado para el diseño de la aplicación ha sido de tipo Modelo – Vista – Controlador (MVC). Este tipo de arquitectura usa un patrón, que a su vez define la organización independiente de los objetos del negocio (Modelo), la interfaz con el usuario u otro sistema (Vista) y el controlador del flujo de trabajo de la aplicación (Controlador) [1][2].

Por tanto, podemos decir que en la arquitectura MVC:

- ◆ **Modelo:** Contiene el núcleo de la funcionalidad de la aplicación, encapsula el estado de la misma y es independiente del controlador y la vista.
- ◆ **Vista:** Es la encargada de la presentación del modelo, pudiendo acceder al modelo pero nunca cambiando su estado y pudiendo ser notificada cuando hay un cambio en el estado del modelo.
- ◆ **Controlador:** Reacciona a las peticiones que realiza el cliente, ejecutando para cada una de ellas las acciones adecuadas y creando el modelo pertinente.

A continuación, se presentas el diseño de los componentes de la arquitectura. El orden utilizado facilitará la comprensión del diseño realizado.

4.3.2 VISTA

El cliente nos ha indicado claramente como quiere la distribución de la información en este gestor de contenidos, como se mencionó anteriormente, hemos tomado como referencia los bocetos que se comentan en la propuesta el proyecto aprobada por el cliente:

- La distribución de la información a lo largo de la aplicación debe seguir un diseño uniforme: se dividirá en *tres zonas*, la información mostrada en la *zona central* irá variando en base a las opciones de menús seleccionadas. Y la zona de menú será distinta en la página principal de la aplicación con respecto a la página principal de usuarios registrados, ya que pueden realizar otras operaciones.
- Para la realización de las zonas, utilizaremos “capas” en “HTML”. Las capas se pueden definir como áreas diferenciadas que se pueden incluir en una página. Los atributos de una capa (posición, visibilidad, etc.), como los de cualquier otro elemento HTML, se definen dentro de una hoja de estilo [8].

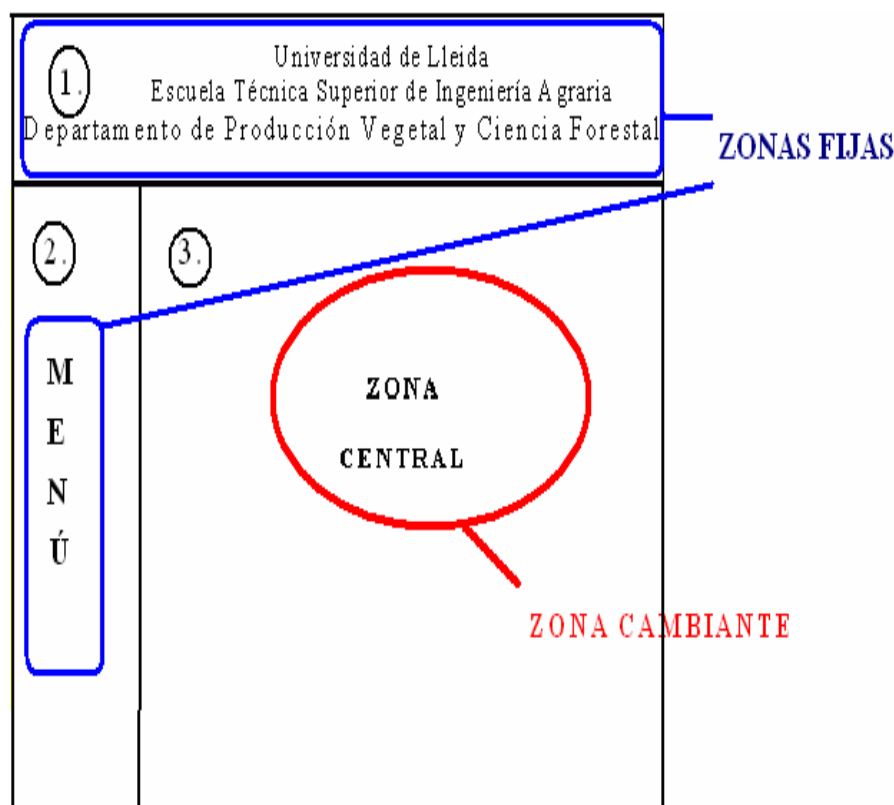


Figura 4-4: Estructura principal de las páginas

- La página principal de la aplicación contendrá información comentada anteriormente, donde **existirán dos tipos de páginas** de un mismo sitio bien diferenciadas, los *públicos* y el acceso a zona *privada*:

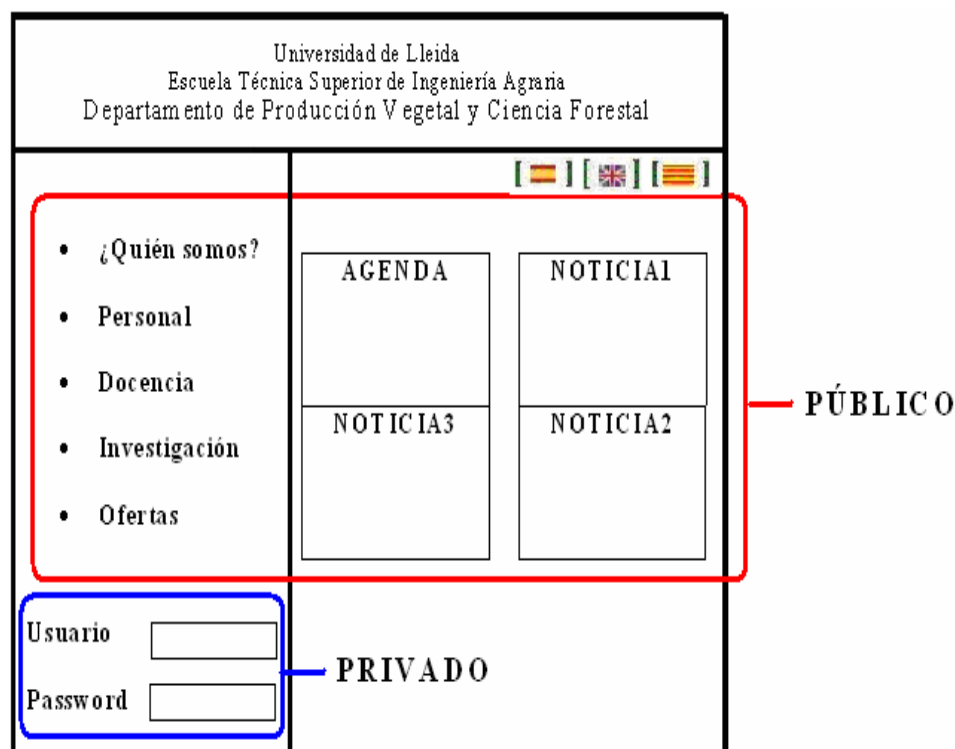
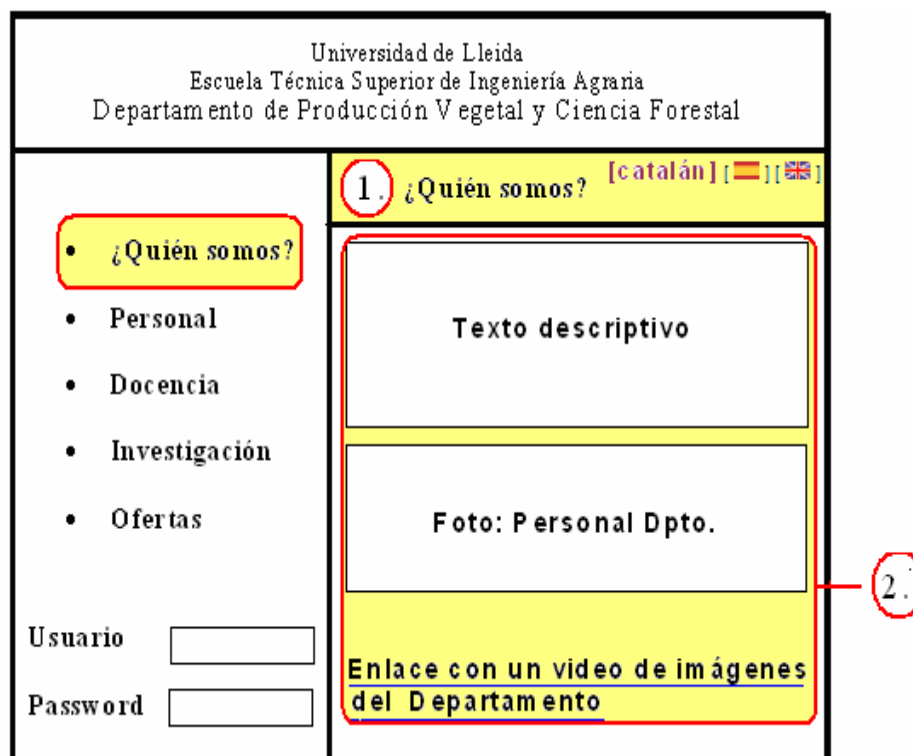


Figura 4-5: Boceto de página principal

- La **información central** variará, pero siempre mantendremos como título principal lo que se ha seleccionado en el menú. Por ejemplo, para mostrar el apartado *¿Quiénes somos?*:



Universidad de Lleida
Escuela Técnica Superior de Ingeniería Agraria
Departamento de Producción Vegetal y Ciencia Forestal

1. ¿Quiénes somos? [catalán] [es] [en]

- ¿Quiénes somos?
- Personal
- Docencia
- Investigación
- Ofertas

Usuario

Password

2. Foto: Personal Dpto.

Enlace con un video de imágenes del Departamento

Figura 4-6: Boceto de página de ¿Quiénes somos?

La zona marcada con 1. contiene el “Título” principal de la información que se muestra en la zona 2. En este caso, por petición del usuario, la información central está compuesta por: un texto descriptivo, foto del grupo y un enlace a videos.

Por otro lado, las páginas que se muestran cuando un usuario decide iniciar su sesión, mantienen el mismo formato que las páginas de los usuarios no registrados. Exceptuando que contendrán:

- **Operaciones** distintas en el menú.
- Se la dará la **bienvenida** al usuario de la sesión
- **No aparece la** opción de seleccionar **idiomas** porque el idioma ha sido seleccionado previamente, en la página principal del portal Web.
- El usuario puede **salir** de la sesión y volver la página de inicio del portal Web.
- Siempre podrá ir a **la página principal** de usuarios registrados.

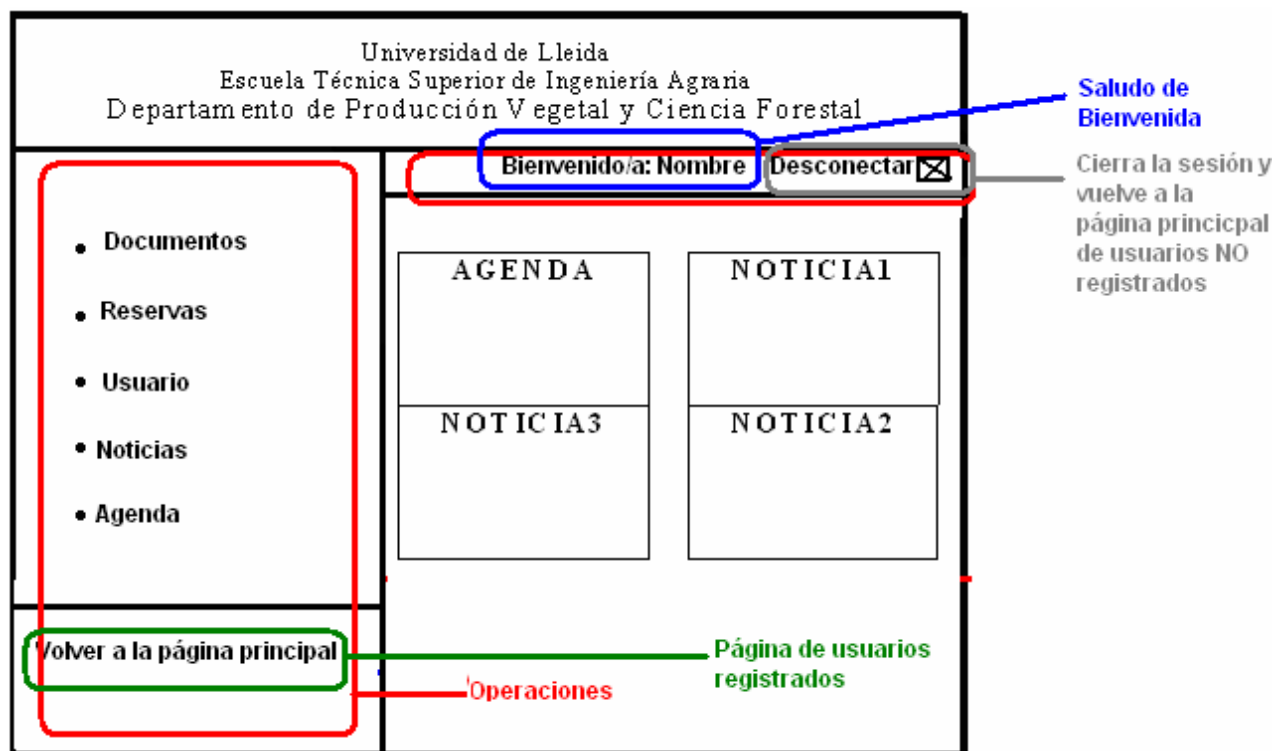


Figura 4-7: Página principal usuarios registrados

Al igual que la página principal de la aplicación, esta página principal de usuarios registrados contiene la información de las noticias y de la agenda, pero privada.

Para el resto de páginas mostrara la información con el mismo diseño de página, en el que varía la información central dependiendo de la operación ejecutada.

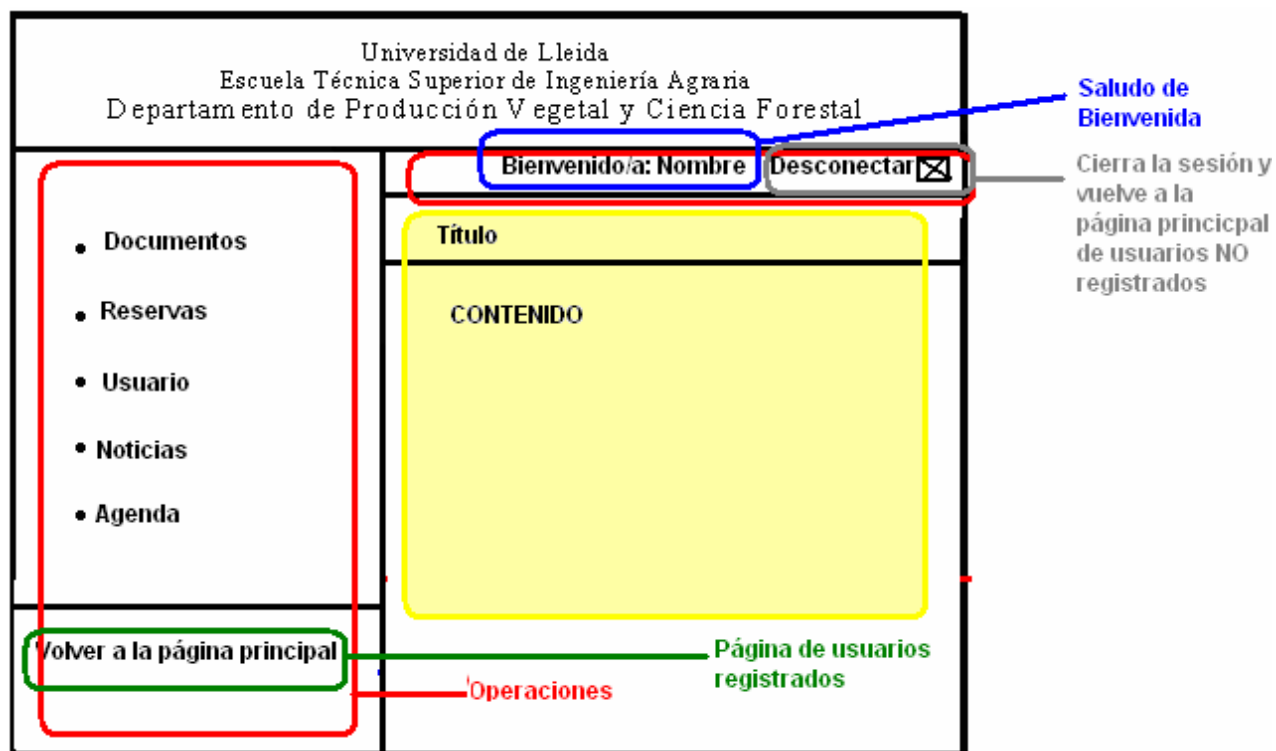


Figura 4-8: Diseño de páginas usuario registrado



4.3.3 MODELO: DIAGRAMA ENTIDAD-RELACIÓN (E/R)

En este apartado se muestra el diseño de la Base de Datos desarrollada para poder satisfacer las necesidades contempladas en los requisitos. Se comienza nombrando las entidades que necesitamos para poder gestionar la información del Portal Web y poder satisfacer los requisitos que se han comentado en apartados anteriores. En esta introducción no detallamos los atributos que contienen cada entidad ya que se muestran a continuación en un diagrama.

- Para almacenar la información que aparece en el portal Web se han necesitado entidades que recojan las características asociadas a cada concepto que se quiere ver, borrar, modificar o añadir en el Portal Web:
 - Como en la información que aparece en la **página de inicio** del portal y **página inicial de usuarios registrados** es la agenda y las noticias, se han definido dos entidades:
 - Agenda
 - Noticia
 - La **docencia** que imparte el departamento son las licenciaturas, master y cursos de especialización, así que para almacenar esta información se definen tres entidades, pero también se necesita la entidad asignatura ya que todas las licenciaturas, masters y cursos de especialización tienen más de una asignatura asociada:
 - Asignatura.
 - Licenciatura
 - CursosEspecialización
 - Master
 - La información de **investigación** que gestiona el departamento son las tesis, transferencias tecnológicas y los grupos de investigación, por lo que se crean tres entidades:
 - Tesis
 - TransferenciaTecnológica
 - GrupoInvestigación
 - La información de las **ofertas** del departamento son tres: las becas, los proyectos finales de carrera y los trabajos dirigidos.



Así que se definen tres entidades para que recogen las características de estos conceptos:

- Beca
- Proyecto Final de Carrera (PFC)
- TrabajoDirigido
- Para almacenar los **documentos** que pueden incluir los miembros del departamento en el portal Web:
 - Documento
- Para mostrar el **personal** que pertenece al departamento, se necesita una entidad que sea usuario y recoja la información asociada. A parte se define una entidad para la gestión de funcionalidades dentro del personal del departamento, en este caso, sólo los usuarios administradores pueden realizar *todas* las operaciones de “usuarios registrados” en el portal, pero puede que en ampliaciones futuras se diferencien más funcionalidades y más perfiles dentro de los “usuarios registrados”, por este motivo se añade esta entidad TipoUsuario:
 - Usuario
 - TipoUsuario: esta entidad sirve para diferenciar el perfil de los usuarios registrados, por ejemplo, un usuario registrado como administrador.
- Para la gestión de las **reservas** realizadas por el personal del departamento, se definen cuatro entidades correspondientes a los objetos que se pueden reservar:
 - Coche
 - Cañon
 - Laboratorio
 - Ordenador
- Una vez que hemos definido las entidades para representar los conceptos que se quieren gestionar en el portal Web, se requieren entidades para **relacionar** al menos dos entidades de las citadas anteriormente y características que las unen, por este motivo se definen estas entidades:

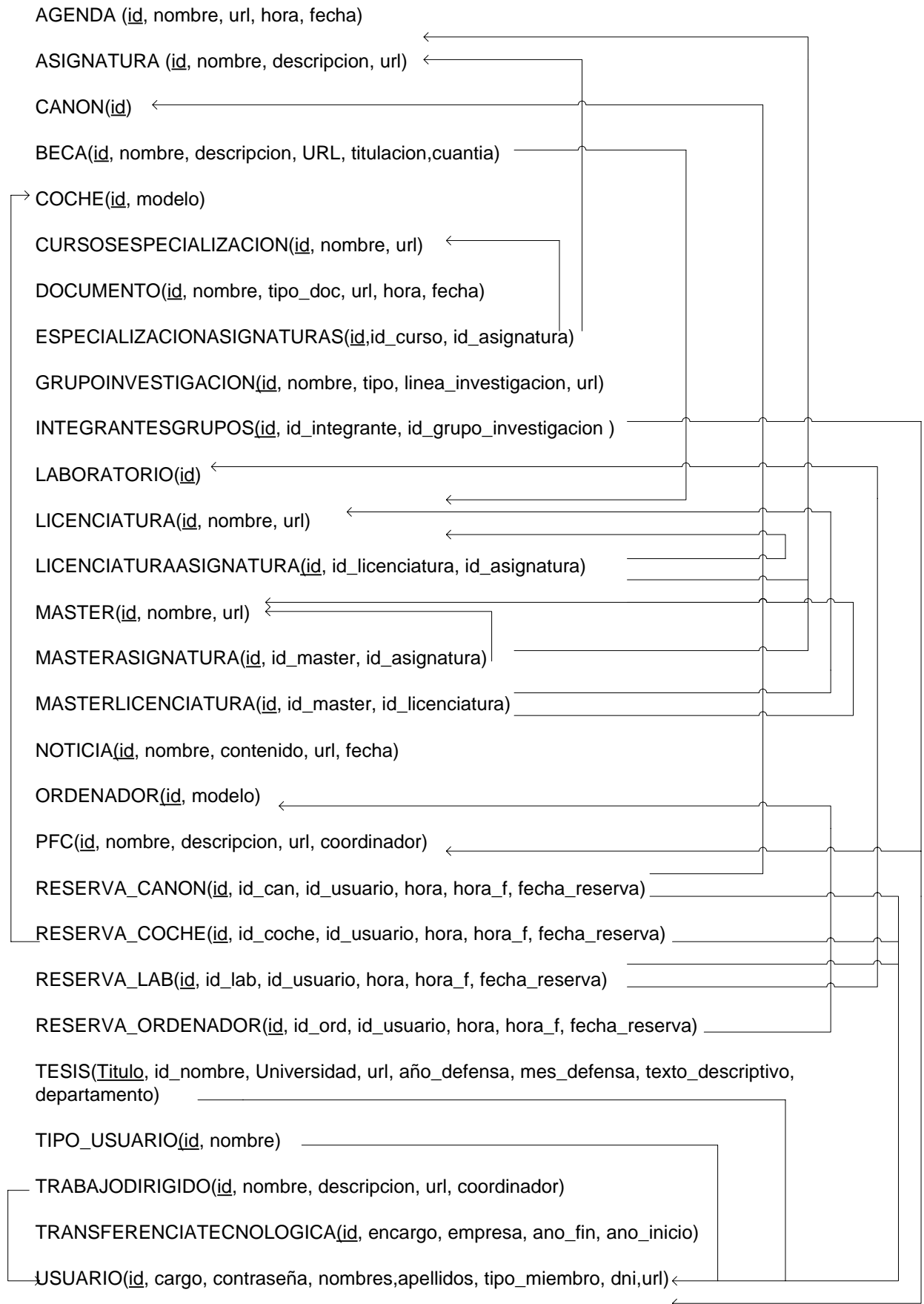


- MasterAsignatura: Relaciona un master con una asignatura.
- MasterLicenciatura: Relaciona un master con una licenciatura.
- LicenciaturaAsignatura: Relaciona una licenciatura con una asignatura.
- EspecializacionAsignatura: Relaciona un curso de especialización con una asignatura.
- IntegrantesGrupo: Relaciona un miembro del departamento con un grupo de investigación.
- Reserva_Coche: Relaciona un miembro del departamento que reserva un coche.
- Reserva_Cañon: Relaciona un miembro del departamento que reserva un cañón.
- Reserva_Lab: Relaciona un miembro del departamento que reserva un laboratorio.
- Reserva_Ordenador: Relaciona un miembro del departamento que reserva un ordenador.

Una vez comentadas las entidades que se utilizan y el concepto que representan, se muestra un diagrama entidad-relación con todos los atributos que definen estas entidades.

Para hacer más sencilla la univocidad de las entidades se decide utilizar como **clave primaria** en todas un rango numérico, definido como “id”, identificador de la entidad, excepto en Tesis que se considera cómo clave unívoca y primaria el título de la misma, ya que no puede existir dos tesis con el mismo título.

Respecto al resto de los atributos, todos los campos de las entidades son obligatorios y respecto al atributo “dni” de la entidad “usuario”, también es una clave unívoca, ya que una persona no puede tener un “dni” repetido.





4.3.4 CONTROLADOR

Tomando en cuenta que se utiliza para la implementación el framework de Struts, los detalles de diseño del mismo se podrán apreciar en el apartado de 4.4 *Implementación*.



4.4 IMPLEMENTACIÓN DE LA APLICACIÓN

Una vez comentado el diseño de la aplicación, pasamos a la fase de implementación. Este apartado se explica en base a los paquetes de la arquitectura elegida (MVC). Comenzaremos con el componente Vista, después el Controlador y para concluir el Modelo.

Al final del apartado se añaden unos diagramas de secuencia que muestran la interacción entre los tres paquetes de la arquitectura.

4.4.1 VISTA

Tomando en consideración el diseño previamente presentado, para la realización de las vistas, se ha utilizado:

- ◆ **JavaServer Pages (JSP):** es una tecnología Java que permite generar contenido dinámico para Web, en forma de documentos HTML, XML o de otro tipo [3].

Para la creación de las JSP's se utiliza el concepto “capa”, que es un recipiente rectangular para contenido HTML que puede situarse en una ubicación exacta en la ventana del navegador. Las capas pueden contener una amplia variedad de elementos: texto, imágenes, tablas e incluso otras capas. Cualquier elemento que pueda ubicarse en un documento HTML puede ubicarse en una capa. Así pues, se diseña cada JSP compuesta por tres capas:

- Capa S: Capa superior, que contiene el encabezado y las opciones de selección de idiomas (en las páginas de usuarios no registrados) o la bienvenida de usuario más la opción de desconectar (en las páginas de usuarios registrados).
- Capa Izquierda: corresponde a las opciones de menú. Esta capa es distinta en las vistas asociadas a los usuarios registrados respecto a las de los usuarios no registrados, ya que el menú varía entre unas y otras.



- Capa Central: Contiene la parte dinámica de la aplicación, ya que dependerá de la opción seleccionada por el usuario.

Otro elemento que se ha utilizado para la elaboración de las páginas, son las hojas de estilo CSS (Cascading Style Sheets) ya que permiten definir el modo en que se dispone el texto en las páginas Web. El término “cascading” se refiere a la secuencia ordenada de estilos. Un estilo es un grupo de atributos identificado por un sólo nombre. Los estilos en los documentos HTML ofrecen un gran control sobre el formato de texto. La ventaja de utilizar estilos reside en que cuando se realiza un cambio en un atributo del estilo, todo el texto controlado por ese estilo cambiará automáticamente de formato, por lo que da flexibilidad a la apariencia de la página de manera sencilla. Se han utilizado hojas de estilo para representar:

- Títulos del menú
- Títulos del contenido principal de cada página
- Subtítulos de las páginas
- Contenido de las páginas
- Contenido de los mensajes de error

La página principal de la aplicación se muestra en el idioma español. Si el usuario decide cambiar de idioma, al seleccionar el idioma, se modificará un atributo de sesión “Idioma” y como consecuencia se mostrará toda la información en la lengua seleccionada.

Para satisfacer el requisito de que la aplicación debe mostrar la información en tres idiomas, en la capa Izquierda y en la Central de las páginas, se tiene toda la información que es estática traducida en los tres, de tal manera que mostrará la información estática dependiendo del atributo de sesión “Idioma”. De esta manera, nos evitamos el tener que triplicar las vistas de la aplicación. Pero para la página principal, no se ha podido evitar este hecho, así que se tiene:

- La vista principal de la aplicación en español, “index2.jsp”.

- La vista principal de la aplicación en inglés, “index2_I.jsp”.
- La vista principal de la aplicación en catalán, “index2_C.jsp”.

Mostramos el diagrama de clases correspondiente a este paquete.

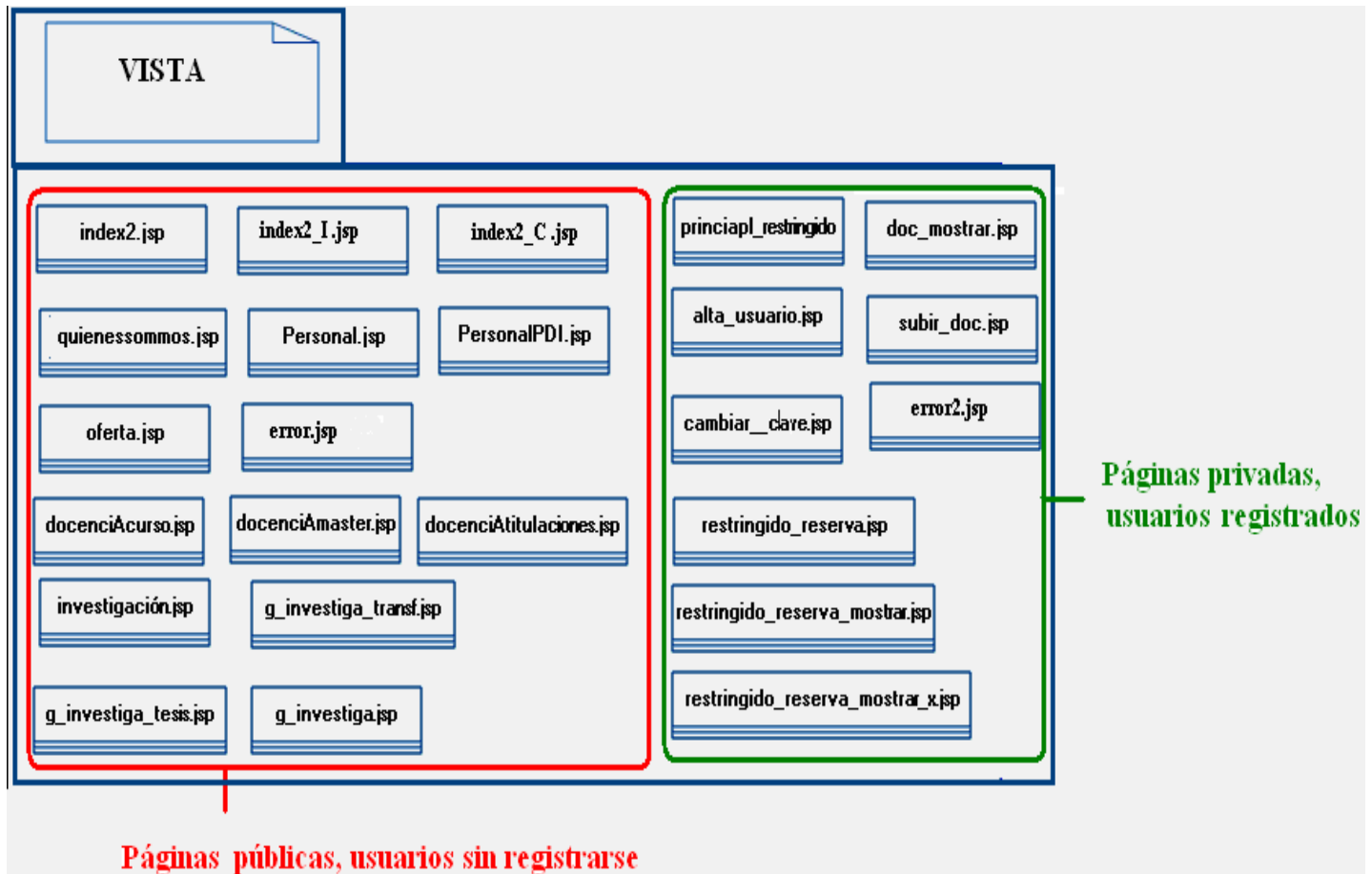


Figura 4-9: Conjunto de JSP's del paquete Vista

Cuando ejecutamos una operación, pueden ocurrir las siguientes posibilidades:

1. En las operaciones de consulta, le lleva a la página que solicita el usuario. Por ejemplo, cuando se selecciona la opción de menú “¿quienes somos?”, se muestra la página JSP “quienessomos.jsp”.
2. En las operaciones que implican una modificación, borrado o inserción, una vez realizadas, se muestra la lista de objetos asociadas, es decir, si por ejemplo modificamos una reserva, si se ha realizado con éxito la operación nos mostrará la lista de las reservas para que podamos verificar la modificación. En el caso de error, se mostrará una vista de



error, “error.jsp” o “error_2.jsp”, dependiendo si la operación de error es de un usuario registrado en la aplicación o no.

3. Vistas de error. Se han realizado 2 JSP's, “error.jsp” y “error_2.jsp”, para mostrar los posibles errores al usuario, el contenido que se muestra en la página de error será un mensaje que varía dependiendo del error. La JSP “error.jsp” contiene el menú principal de operaciones de la aplicación para poder continuar con otras operaciones y la JSP “error_2.jsp” el menú de la aplicación de usuarios registrados, para facilitar al usuario registrado no tener que acceder de nuevo al sistema de validación.



4.4.2 CONTROLADOR

Reacciona a las peticiones que realiza el cliente, ejecutando para cada una de ellas las acciones adecuadas y creando el modelo pertinente. Para representar el papel de controlador se utilizan los *servlets*, los cuales pueden ser invocados desde páginas JSP o desde otros *servlets* con unos parámetros de entrada (datos de un formulario...), para realizar las acciones (llamadas al modelo u otras operaciones) y redirigir los resultados a una página (JSP). Para esta redirección se ha utilizado **STRUTS**, ya que permiten tener en un fichero todas las posibles redirecciones en función del resultado del *servlet*. El framework de Struts está formado por los siguientes componentes [1]:

- ◆ **Action:** Paquete que contiene todas las clases que redirigen una llamada a su correspondiente ejecución, según la acción que se vaya a llevar a cabo.
- ◆ **Form:** Corresponde a los objetos ActionForms, los cuales facilitan el transporte de datos entre las capas de la aplicación. Son utilizados por las clases para capturar los datos procedentes de los formularios que se encuentren en las JSP's y enviárselos al objeto Action correspondiente. Así pues, se crean Form para las situaciones en las que el usuario introduce información escrita en la aplicación, es decir para cuando rellena un formulario o introduce datos.
- ◆ **Fichero de configuración struts-config.xml:** fichero de configuración en donde se registran y configuran los distintos componentes de Struts que van a ser utilizados por la aplicación.

A continuación mostramos el diagrama de clases correspondiente a los Actions y ActionForms desarrollados:

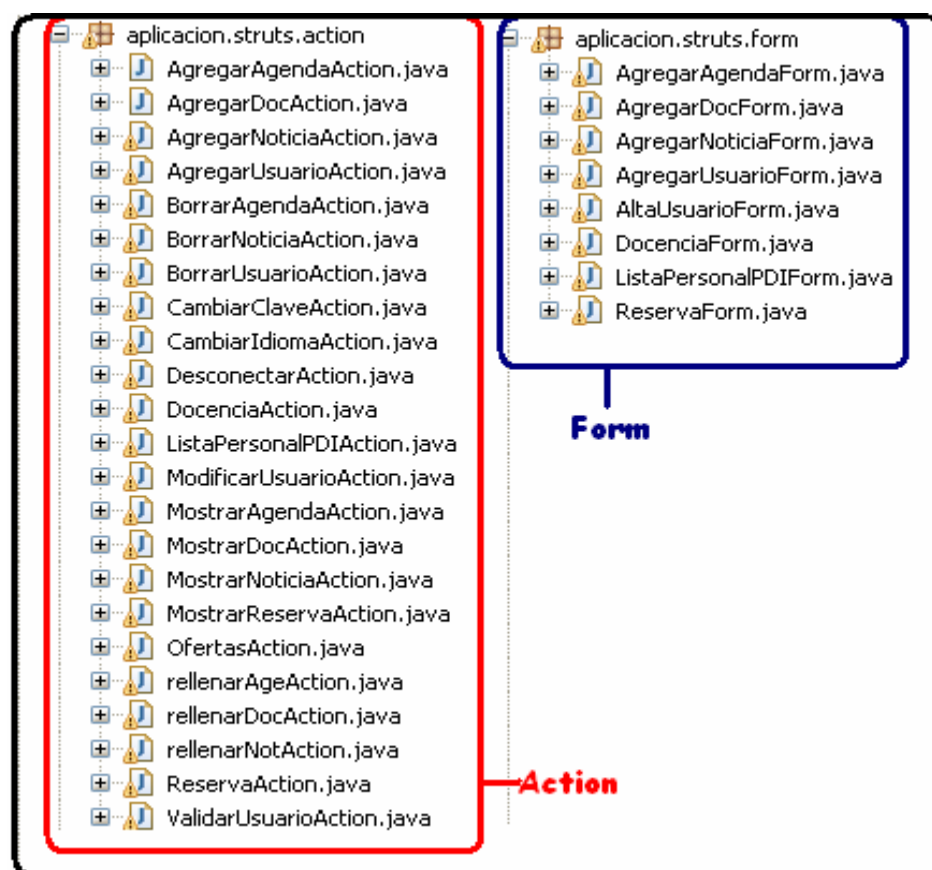


Figura 4-10: Clases que implementan Struts

En el fichero de configuración **struts-config.xml** deben de estar definidos todos los Form y Action que se encontraban en la “Figura 4-10: Struts”. Este fichero tiene una serie de “tags” para poder definir los Action y los ActionForms:

- Todos los Form se definen entre los tags `<form-beans>` `</form-beans>`.
- Entre estos tags `<action-mappings>` `</action-mappings>` se define una asociación entre tipos de petición y objetos Action, además de objetos forward locales. Entre estos tags, se incluyen otros llamados `<action>``</action>`, en ellos se definen los parámetros asociados a cada acción. A continuación procedemos a explicar cada uno de estos parámetros:



- **Path:** Representa el Servlet path que debe ser utilizado en la URL de la petición para que ActionServlet derive la petición a este objeto Action.
- **Name:** nombre del objeto ActionForm.
- **Type:** Nombre cualificado de la subclase Action.
- **Scope:** Ámbito de utilización del objeto ActionForm. Su valor puede ser a nivel de petición (*request*), a nivel de sesión (*session*) o a nivel de aplicación (*application*).
- **Elementos** `<forward></forward>`: Estos “tags” representan las posibles vistas que puede encaminar el usuario tras la ejecución de la acción, indicándose en cada uno de ellos la asociación virtual de la vista y la dirección física. Disponen de dos elementos:
 - **Name:** Nombre lógico o virtual asociado a la vista. Este valor es el utilizado por el método *findForward()* de ActionMapping para crear el objeto ActionForward asociado a la vista a la que se tiene que dirigir al usuario.
 - **Path:** contiene la URL relativa a la vista.

Mostramos parte del fichero para que se pueda comprender mejor lo explicado previamente:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts-config PUBLIC "-//Apache Software Foundation//DTD Struts Configuration 1.2//EN"
"http://struts.apache.org/dtds/struts-config_1_2.dtd">

<struts-config>
  <data-sources />

  <form-beans>
    <form-bean name="agregarUsuarioForm"
      type="aplicacion.struts.form.AgregarUsuarioForm" />
    <form-bean name="listaPersonalPDIForm"
      type="aplicacion.struts.form.ListaPersonalPDIForm" />
    <form-bean name="docenciaForm" .....
      type="aplicacion.struts.form.DocenciaForm" />
    <form-bean name="reservaForm"
      type="aplicacion.struts.form.ReservaForm" />
    [...]
  </form-beans>

  <global-exceptions />
  <global-forwards />
</struts-config>
```




```
<action-mappings>
  <action
    attribute="listaPersonalPDIForm"
    name="listaPersonalPDIForm"
    path="/PersonalPDI"
    scope="request"
    type="aplicacion.struts.action.ListaPersonalPDIAction"
  >
    <forward
      name="exito"
      path="/PersonalPDI.jsp"
    />
    <forward
      name="error"
      path="/error.jsp"/>
    <forward
      name="inicio"
      path="/index.jsp"/>
  </action>
  [...]
</action-mappings>

<controller processorClass="org.apache.struts.tiles.TilesRequestProcessor"/>
  <message-resources parameter="aplicacion.struts.ApplicationResources" />
<plug-in className="org.apache.struts.tiles.TilesPlugin" >
  <set-property property="definitions-config" value="/WEB-INF/tiles-defs.xml" />
  <set-property property="moduleAware" value="true" />
</plug-in>

</struts-config>
```

A continuación vamos a entrar en más detalle en las funcionalidades que realizan cada uno de estos Action. No necesariamente debe haber un Form por cada Action. Se crean Form para las situaciones en las que el usuario introduce información escrita en la aplicación, es decir para cuando rellena un formulario o introduce datos.

Repasando las funcionalidades que debe realizar la aplicación se resumen en:

- Consultar información en la Base de Datos y mostrarla:
 - i. Personal del departamento: PDI, PAS, Becarios
 - ii. Docencia: Titulaciones, Masters y cursos.
 - iii. Ofertas: Proyectos Fin de Carrera (PFC), Becas, Trabajos dirigidos.
 - iv. Investigación: grupos de investigación, Tesis y transferencia tecnológica.
 - v. Documentos: actas, consejos
 - vi. Reservas: coches, ordenadores, laboratorios y cañones.



- vii. Noticias y agenda.
- viii. Validar usuario del departamento.
- Añadir información en la Base de Datos
 - i. Reservas: de coche, cañón, laboratorio y ordenador.
 - ii. Documentos: actas o congresos
 - iii. Usuarios del sistema
 - iv. Noticias y agenda
- Modificar información en la Base de Datos
 - i. Reservas: de coche, cañón, laboratorio y ordenador.
 - ii. Usuarios del sistema
 - iii. Noticias y agenda
- Borrar información en la Base de Datos
 - i. Reservas: de coche, cañón, laboratorio y ordenador.
 - ii. Usuarios del sistema
 - iii. Noticias y agenda
- Desconectar.

Casi todas las acciones de añadir información a la Base de datos tienen un *form* asociado. A continuación se muestra en la siguiente tabla la relación entre las funcionalidades a desarrollar y los “Actions” y “Forms” implementados para satisfacerlas.



FUNCIONALIDAD	ACTION	FORM
CONSULTAR BD		
Consultar personal de la universidad: PDI, PAS, Becarios	ListaPersonalPDI.Action.java	ListaPersonalPDIForm.java
Consultar docencia: Titulaciones, masters y cursos.	DocenciaAction.java	DocenciaForm.java
Consultar ofertas: PFC's, Becas, Trabajos dirigidos.	OfertasAction.java	-----
Consultar investigación: grupos de investigación, Tesis y transferencia tecnológica.	Se introduce código en las JSP's: <ul style="list-style-type: none"> • g_investigacion.jsp • g_investiga.jsp • g_investiga_tesis.jsp • g_investiga_transf.jsp 	-----
Consultar documentos: actas, Consejos	MostrarDocAction.java	-----
Consultar reservas: coches, ordenadores, laboratorios y cañones.	MostrarReservaAction.java	-----
Consultar noticias y agenda.	Se introduce código en la JSP: <ul style="list-style-type: none"> • index_2.jsp • principal_restringido.jsp 	-----
Consultar validar usuario del departamento.	ValidarUsuarioAction.java	AgregarUsuarioForm.java
AÑADIR INFORMACIÓN A LA BD		



Reservas: de coche, cañon, laboratorio y cañon.	ReservaAction.java	ReservaForm.java
Documentos:actas o congresos	AgregarDocAction.java	AgregarDocForm.java
Usuarios	AgregarUsuarioAction.java	AltaUsuarioForm.java
Noticia	AgregarNoticiaAction.java	AgregarNoticiaForm.java
Agenda	AgregarAgendaAction.java	AgregarAgendaForm.java
MODIFICAR INFORMACIÓN EN LA BD		
Reservas: de coche, cañon, laboratorio y cañon.	ReservaAction.java	ReservaForm.java
Documentos:actas o congresos	AgregarDocAction.java RellenarDocAction.java	AgregarDocForm.java
Usuarios	AgregarUsuarioAction.java	AltaUsuarioForm.java
Noticia	AgregarNoticiaAction.java RellenarNotAction.java	AgregarNoticiaForm.java
Agenda	AgregarAgendaAction.java RellenarAgeAction.java	AgregarAgendaForm.java
BORRAR INFORMACIÓN DE LA BD		
Usuarios	BorrarUsuarioAction.java	-----
Noticia	BorrarNoticiaAction.java	-----
Agenda	BorrarAgendaAction.java	-----
Reservas	BorrarReservaAction.java	-----
DESCONECTAR		
Desconectar	DesconectarAction.java	-----



Como se puede apreciar, prácticamente todas las acciones de la aplicación se corresponden con una clase *Action* y los *Form* están en todas las acciones de modificación de datos de la Base de Datos. Vuelvo a comentar que estas clases **son las mismas** para el resto de idiomas.

4.4.3 MODELO

Y por último, en lo que al modelo se refiere, podemos establecer que está formado por los siguientes componentes:

- ◆ **Hibernate:** Paquete en el que se engloban las clases generadas a partir del Script de la base de datos realizado. Así, dentro de este paquete existe una clase para cada una de las tablas de la base de datos, con el fin de poder obtener la información almacenada en ésta.

La Base de Datos está desarrollada para contemplar tres idiomas, por lo que tenemos las tablas para los tres idiomas. Excepto la información relativa a *usuario*, *reservas* y *objetos a reservar* (coche, cañón, laboratorio y ordenador). que es la misma en los tres.

- ◆ **Servicios:** Paquete que contiene un conjunto de clases que gestionan los datos extraídos de la Base de Datos.

Se define la clase “*GestionServicios.java*” que contiene al menos un método por cada acción a realizar de la aplicación, y de este modo ofrecer los datos necesarios para cada una de las funcionalidades requeridas para la aplicación. Estos métodos son utilizados por las clases *Actions* dentro de sus métodos *executed()* para devolver el resultado de la petición del usuario.

Para acceder a los datos en inglés y en catalán, se definen dos clases más que son una copia de “*GestionServicios.java*”: “*GestionServiciosI.java*” y “*GestionServiciosC.java*”.

- ◆ **Junits:** Este paquete contiene un conjunto de pruebas unitarias para verificar el correcto funcionamiento de los métodos implementados para la manipulación de datos de la Base de Datos. Es decir, existe un



método por cada método implantado en la clase “*GestionServicios*” que comprueba su funcionamiento. De esta manera nos aseguramos que el resultado de la ejecución de las clases *Actions* no debe fallar por los métodos de manipulación de datos. Conseguimos depurar los fallos de manera más rápida y segura.

Como hemos comentado anteriormente, la información en el modelo esta triplicada. Pero las clases son iguales, y se nombran con el mismo nombre más una “*I*” o “*C*” al final. Por ejemplo:

Agenda.java (español) - AgendaI. java (inglés) -AgendaC. java (catalán)

Excepto las clases que son únicas para los tres idiomas que son:

- Usuario.java → AbstractUsuario.java
- ReservaOrdenada.java → AbstractReservaOrdena.java
- ReservaCanon.java → AbstractReservaCanon.java
- ReservaLaboratorio.java → AbstractReservaLaboratorio.java
- ReservaCoche.java → AbstractReservaCoche.java
- Ordenador.java → Ordenador.java
- Canon.java → Canon.java
- Laboratorio.java → Laboratorio.java
- Coche.java → Coche.java

Mostramos a continuación el paquete Hibernate con las clases asociadas al idioma Español:

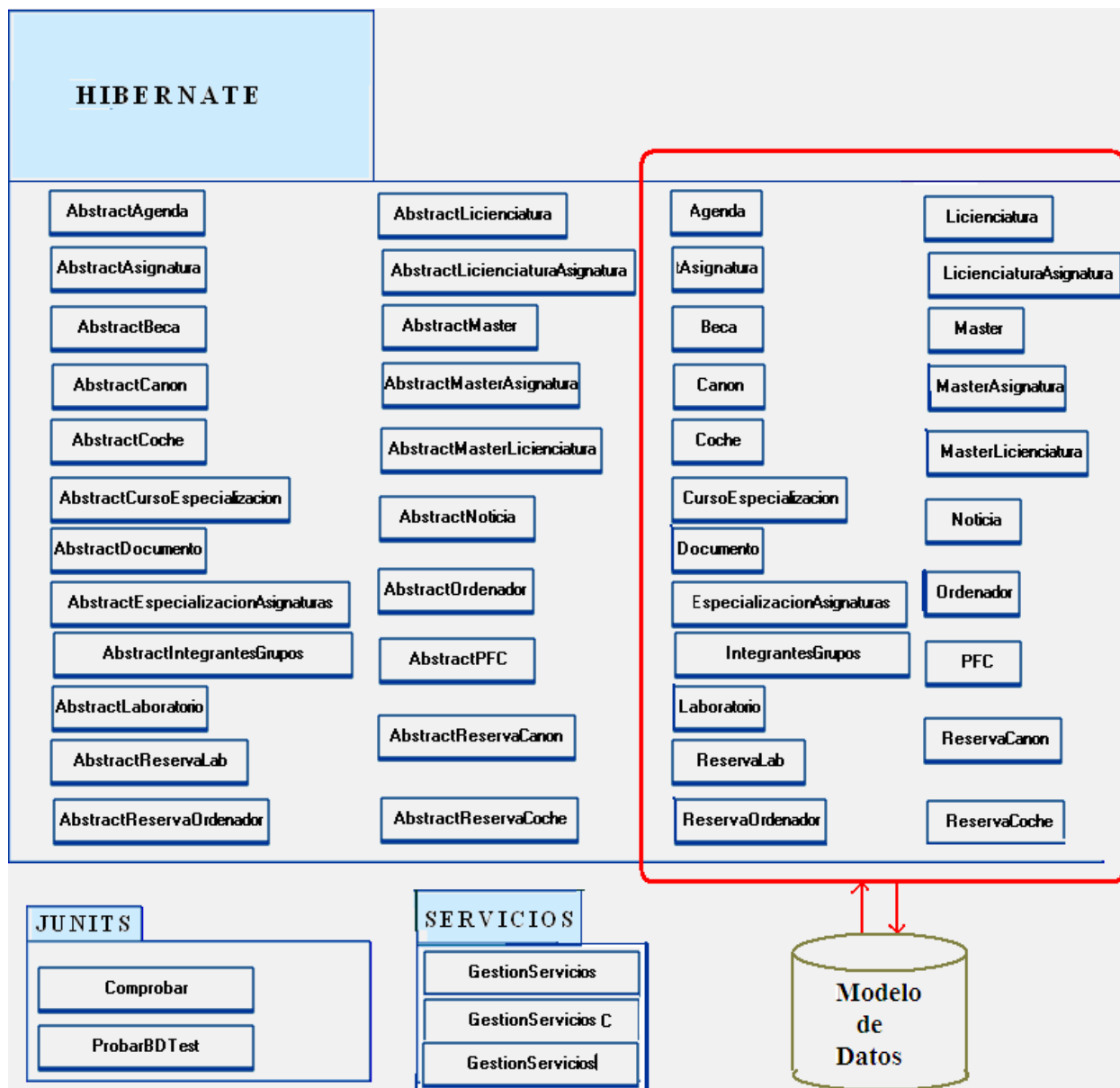


Figura 4-10: Clases del paquete Modelo

En esta figura se puede observar que todas las clases del paquete “Hibernate” que no contiene en su nombre la palabra “Abstract” tiene asociada una clase con el mismo nombre más la palabra “Abstract”, por ejemplo, “Agenda” con “AbstractAgenda”. Las clases que comienzan con *Abstract* son las clases que tienen los métodos para la gestión de los campos de las correspondientes tablas de la base de datos, y las otras, heredan el comportamiento de éstas e incluyen los constructores para crear los objetos de cada una de ellas y permiten ampliar sus funcionalidades. Esto es una ventaja ya que podemos añadir más funcionalidades de las que nos ofrecen las



clases Abstract sin modificar las clases Abstract, ya que provienen del “mapping” del modelo de la Base de Datos que genera Hibernate.

Otro punto importante dentro de Hibernate, es el archivo de configuración que contiene los datos necesarios para **conectarse a la Base de Datos**, éste es *hibernate.cfg.xml* y también en este se incluyen todos los *NombreClase.hbm.xml* que se encargan de mapear cada tabla de la base de datos con un objeto:

<i>NombreClase.hbm.xml</i>	<i>Entidades de la Base de datos</i>
Usuario.hbm.xml	Usuario
Agenda.hbm.xml	Agenda
Canon.hbm.xml	Canon
Licenciatura.hbm.xml	Licenciatura
Tesis.hbm.xml	Tesis
Trabajodirigido.hbm.xml	Trabajodirigido
Especializacionasignaturas.hbm.xml	Especializacionasignaturas
Pfc.hbm.xml	Pfc
Integrantesgrupos.hbm.xml	Integrantesgrupos
Laboratorio.hbm.xml	Laboratorio
Coche.hbm.xml	Coche
Masterlicenciatura.hbm.xml	Masterlicenciatura
Beca.hbm.xml	Beca
Master.hbm.xml	Master
Licenciaturasignatura.hbm.xml	Licenciaturasignatura
ReservaOrdenador.hbm.xml	Reserva_Ordenador
Ordenador.hbm.xml	Ordenador
ReservaLab.hbm.xml	Reserva_Lab
Gruposinvestigacion.hbm.xml	Gruposinvestigacion
Masterasignatura.hbm.xml	Masterasignatura
TipoUsuario.hbm.xml	Tipo_Usuario
Asignatura.hbm.xml	Asignatura
Transferenciatecnologica.hbm.xml	Transferenciatecnologica
Cursoespecializacion.hbm.xml	Cursoespecializacion



ReservaCanon.hbm.xml	Reserva_Canon
Noticia.hbm.xml	Noticia
ReservaCoche.hbm.xml	Reserva_Coche
Documento.hbm.xml	Documento
<i>Para almacenar la información en inglés</i>	
BecaI.hbm.xml	Beca_I
LicenciaturaI.hbm.xml	Licenciatura_I
UsuarioI.hbm.xml	Usuario_I
LicenciaturasignaturaI.hbm.xml	Licenciaturasignatura_I
GruposinvestigacionI.hbm.xml	Gruposinvestigacion_I
TesisI.hbm.xml	Tesis_I
MasterI.hbm.xml	Master_I
IntegrantesgruposI.hbm.xml	Integrantesgrupos_I
TransferenciatecnologicaI.hbm.xml	Transferenciatecnologica_I
EspecializacionasignaturasI.hbm.xml	Especializacionasignaturas_I
DocumentoI.hbm.xml	Documento_I
TrabajodirigidoI.hbm.xml	Trabajodirigido_I
PfcI.hbm.xml	Pfc_I
CursoespecializacionI.hbm.xml	Cursoespecializacion_I
NoticiaI.hbm.xml	Noticia_I
MasterlicenciaturaI.hbm.xml	Masterlicenciatura_I
MasterasignaturaI.hbm.xml	Masterasignatura_I
AsignaturaI.hbm.xml	Asignatura_I
AgendaI.hbm.xml	Agenda_I
<i>Para almacenar la información en catalán</i>	
AgendaC.hbm.xml	Agenda_C
LicenciaturasignaturaC.hbm.xml	Licenciaturasignatura_C
MasterasignaturaC.hbm.xml	Masterasignatura_C
TransferenciatecnologicaC.hbm.xml	Transferenciatecnologica_C
LicenciaturaC.hbm.xml	Licenciatura_C
BecaC.hbm.xml	Beca_C
TesisC.hbm.xml	Tesis_C



AsignaturaC.hbm.xml	Asignatura_C
PfcC.hbm.xml	Pfc_C
IntegrantesgruposC.hbm.xml	Integrantesgrupos_C
EspecializacionasignaturasC.hbm.xml	Especializacionasignaturas_C
CursoespecializacionC.hbm.xml	Cursoespecializacion_C
NoticiaC.hbm.xml	Noticia_C
MasterlicenciaturaC.hbm.xml	Masterlicenciatura_C
TrabajodirigidoC.hbm.xml	Trabajodirigido_C
DocumentoC.hbm.xml	Documento_C
GruposinvestigacionC.hbm.xml	Gruposinvestigacion_C
MasterC.hbm.xml	Master_C

A continuación se muestra parte del fichero de configuración “*hibernate.cfg.xml*” en el que se recuadra las partes que lo componen:

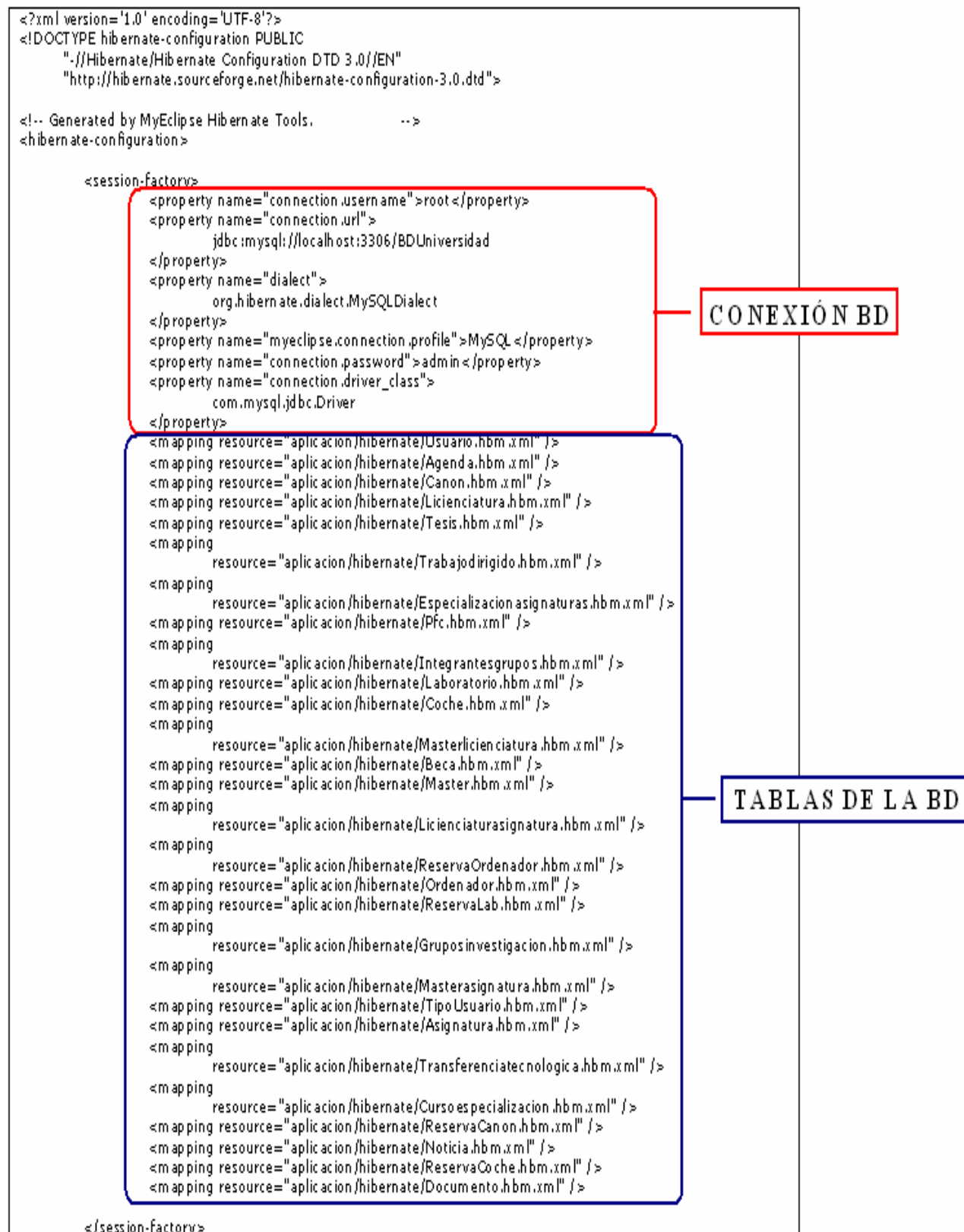


Figura 4-11: Hibernate.cfg.xml

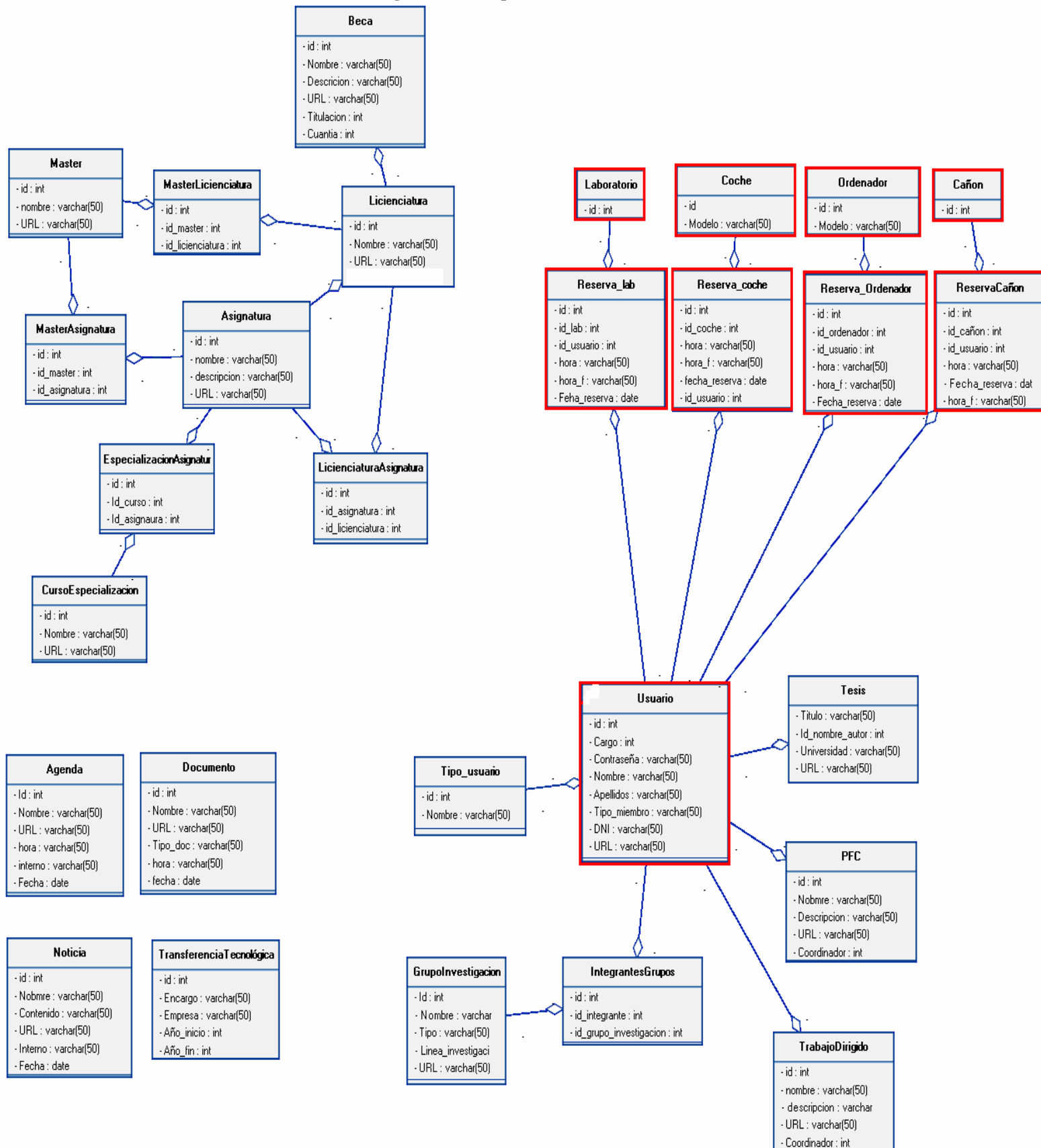


4.4.3.1 PAQUETE HIBERNATE

Vamos a detallar cada uno de los paquetes que se encuentran en el modelo.

Comenzamos por el paquete de Hibernate, el cual contiene una clase por cada tabla de la Base de Datos:

Figura 4-12: Paquete Hibernate



Las clases que están marcadas con rojo, son las que **no** se repiten en tres idiomas, el resto están triplicadas para poder satisfacer la necesidad de soportar tres lenguajes a la vez. Como se ha comentado anteriormente, sólo se le añade al final del nombre una “I” o “C”.

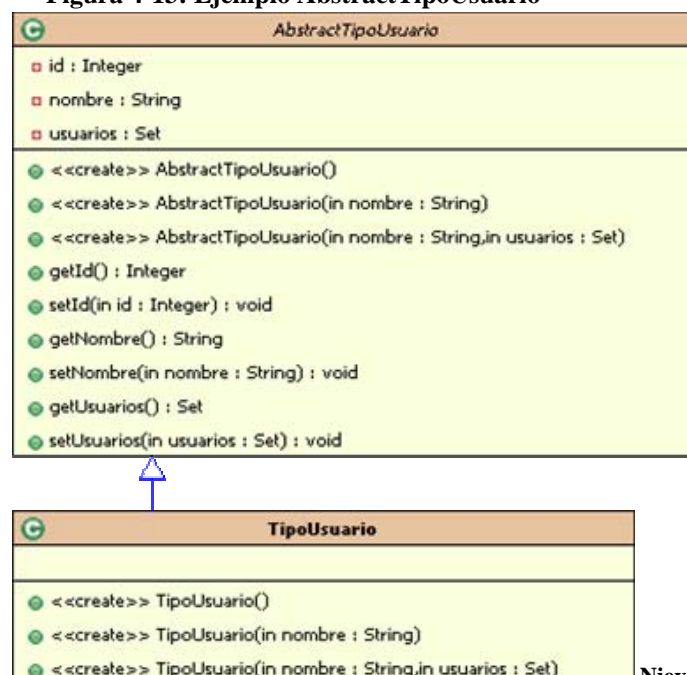
La única diferencia que tenemos del diagrama de la Base de Datos respecto al de clases de Hibernate es que tenemos *Clases Abstract*, en la “Figura 4-15: Paquete Hibernate” no aparecen las clases Abstractas ya que habría tantas como clases mostradas en la figura y no se apreciaría claramente el diagrama. Por este motivo, se ha simplificado la imagen.

Las clases Abstractas contienen la información de las tablas, los métodos para la gestión de los campos asociados a las columnas de las tablas de la Base de Datos, y las *clases no Abstract*, que heredan el comportamiento de éstas incluyendo además los constructores para crear los objetos de cada una de ellas.

Por ejemplo, la tabla *Tipo_Usuario* de la Base de Datos, su correspondencia en el modelo es con la clase *AbstractTipoUsuario* y *TipoUsuario*.

Con estas clases se permite toda la obtención y modificación de los diferentes campos que posee esta tabla, como es obtener o establecer el *identificador* de tipo de usuario, obtener o establecer el *nombre* del tipo de usuario y obtener o establecer los usuarios que pertenecen a un tipo determinado. Así mismo, en la clase *TipoUsuario*, se definen los constructores que permitirán crear un objeto de este tipo para poder trabajar con él.

Figura 4-13: Ejemplo AbstractTipoUsuario



4.4.3.2 PAQUETE SERVICIOS

En este paquete tenemos métodos para agregar, modificar, borrar información de la Base de Datos. Existen tres clases para la gestión de información en base a los idiomas a gestionar, GestionServicios para el castellano, GestionServicios_I para el inglés y GestionServicios_C para el catalán. Los métodos que tienen estas tres clases realizan las mismas funcionalidades exceptuando que dependiendo del idioma se accederá a una clase en inglés, en catalán o en castellano. Por ejemplo, el método “agregarAgenda” en GestionServicios, agregará información de la agenda en castellano, el mismo método en GestionServicios_I, agregará la información de la agenda en inglés y en GestionServicios_C en catalán.

A continuación, se muestra una figura con los métodos que existen en GestionServicios: los métodos desarrollados para agregar, borrar y modificar información de la aplicación:



Figura 4-14: GestionServicios (1)

También se ha tenido que implementar métodos para que muestren la información de la Base de datos:

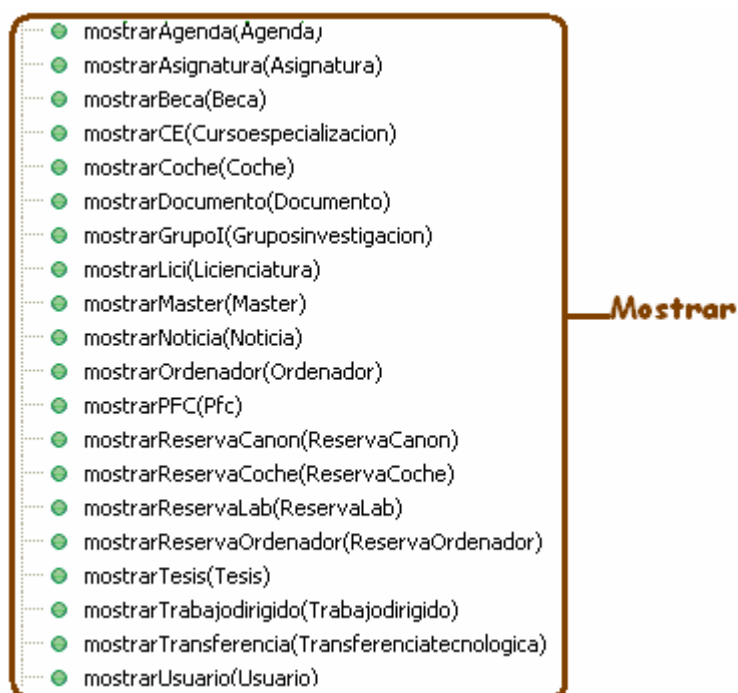


Figura 4-15: GestionServicios(2)

Para la consulta de datos de la Base de Datos, dadas unas condiciones de selección, se ha implementado los métodos “get” y las condiciones de selección se introducen en el método por parámetro. En la siguiente figura se muestran los métodos “get” desarrollados:

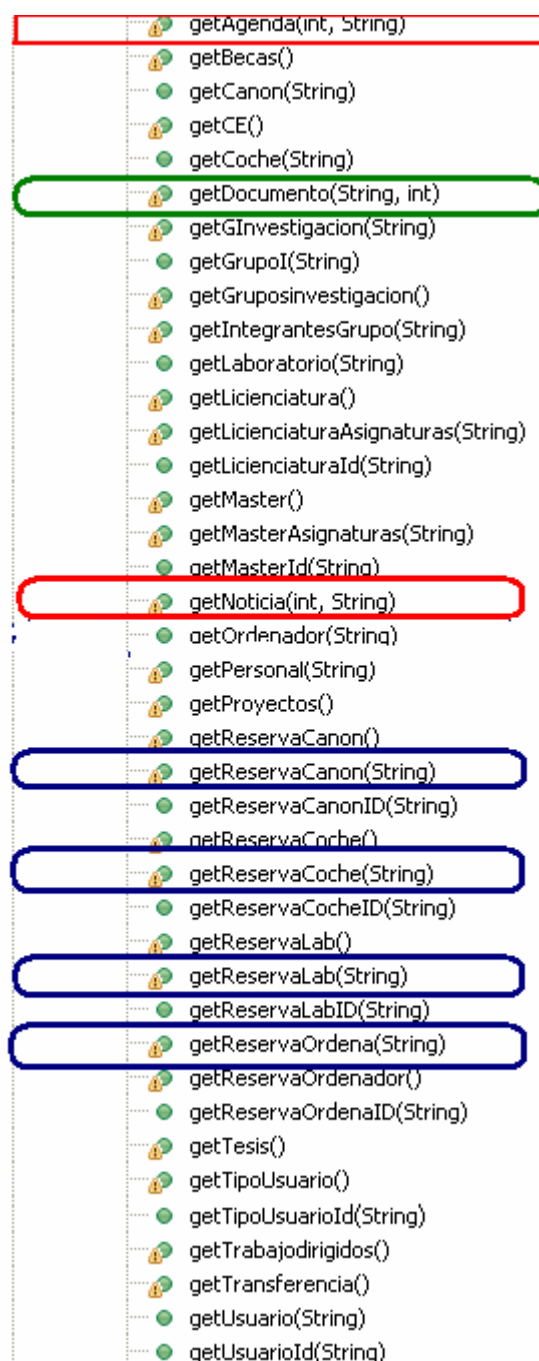


Figura 4-16: GestionServicios(3)

De los métodos mostrados anteriormente, se han subrayado algunos para comentar aspectos de implementación que se han desarrollado con el fin de satisfacer los requisitos:

- Para mostrar la información en la sección de la agenda se utiliza el método *getAgenda(int num_acontecimiento, String interno)*, el cuál selecciona la información de la agenda desde la fecha actual hasta el siguiente mes. Pero está desarrollado para que sólo aparezcan 4 eventos de esta selección ya que



el departamento universitario así lo especificó en los requerimientos. Este número se indica con el parámetro *num_acontecimientos*, si se desea cambiar el *número* de acontecimientos a mostrar, sólo hay que cambiar dicho valor.

- Para mostrar la información en la sección noticias se utiliza el método *getNoticia(int num_acontecimiento, String interno)*, el cuál selecciona las noticias desde la fecha actual hasta el siguiente mes. Pero está desarrollado para que sólo aparezcan 4 eventos de esta selección ya que el departamento universitario así lo especificó en los requerimientos. Este número se indica con el parámetro *num_acontecimientos*, si se desea cambiar el número de acontecimientos a mostrar, sólo hay que cambiar dicho valor.
- Cuando se desarrolla la funcionalidad de mostrar todas las reservas de ordenadores, (*getReservaOrdenador()*), laboratorios (*getReservaLaboratorio()*), cañones (*getReservaCanon()*) o coches (*getReservaCoche()*), se considera que lo más lógico es mostrar las reservas con fecha de reserva a partir de la fecha actual. Se decide que el valor de fecha máxima es hasta un año siguiente, es decir se mostraran todas las reservas que se hayan creado en la aplicación desde la fecha actual hasta el año siguiente (si existen).
- Para mostrar los documentos que el departamento ha insertado en el portal Web se desarrolla el método *getDocumento(String id, int numero)*, el cual muestra como máximo un número predeterminado de documentos, el departamento nos indica que muestren como máximo 6 documentos. Si se desea cambiar el número de documentos acontecimientos a mostrar, sólo hay que cambiar el valor número.

4.4.3.3 PAQUETE JUNITS

El paquete *Junit* esta formado por dos clases: la clase *Comprobar*, que contiene los métodos necesarios para probar los implementados en la clase *GestionServicios*, y la clase *ProbarBDTest* que contiene el *main* para probar los métodos desarrollados en la clase *Comprobar*:



Figura 4-17: Paquete JUnit (1)

- Clase ProbarBDTest.java

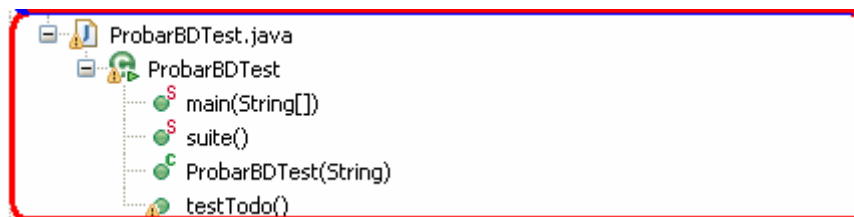


Figura 4-18: Paquete JUnit (2)

- Clase Comprobar.java

Por cada método implementado en las Clases *GestionServicios*, *GestionServiciosI*, *GestionServiciosC* para la manipulación de información de la base de datos, se desarrolla un método “comprobar”, con el cometido de verificar el correcto funcionamiento.



Figura 4-19: Paquete JUnit (3)



4.4.4 DIAGRAMAS DE SECUENCIA

En este apartado se muestran algunos diagramas de secuencia UML para complementar la comprensión del funcionamiento de la aplicación implementada. Así pues, se muestran cuatro:

1. Diagrama de secuencia que representa la tecnología Struts.
2. Diagrama de secuencia correspondiente a una operación de consulta a la Base de datos.
3. Diagrama de secuencia que representa la operación de añadir información de noticias, agenda y documentos en la base de datos.
4. Diagrama de secuencia de la operación de añadir reservas en la Base de Datos.

A continuación se muestra el diagrama de secuencia que sigue la aplicación al interactuar con los paquetes de la arquitectura. Se corresponde con los diagramas de secuencia de la tecnología Struts [1].

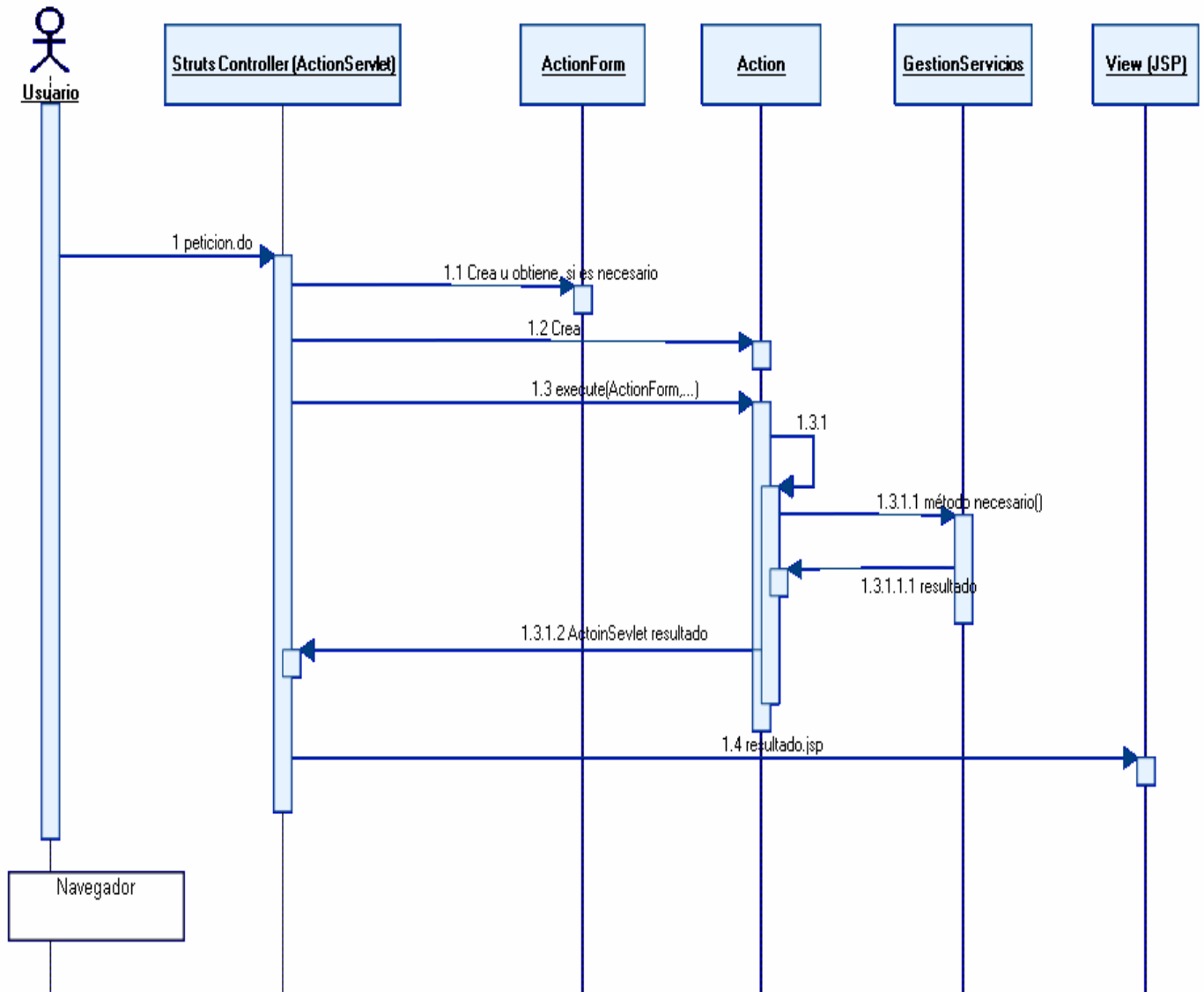


Figura 4-20: Diagrama de secuencia principal

Como hemos comentado en el apartado Funcionamiento de Struts, el diagrama que lo representa es la “Figura 4-22: Diagrama de secuencia principal”. El *Struts Controller(ActionServlet)* utiliza el fichero *struts-config.xml* para:

- Determinar la operación a realizar en base a la *petition.do* recibida.
- Cuando invoca *execute(...)* obtiene como resultado un *ActionServlet* con una dirección virtual encapsulada en el mismo, utiliza el archivo para obtener la dirección real de la página JSP correspondiente.

Cuando se realiza una consulta, se realiza una operación de consulta de la Base de Datos. El diagrama de secuencia que muestran las secuencias de una consulta a la base de datos es:

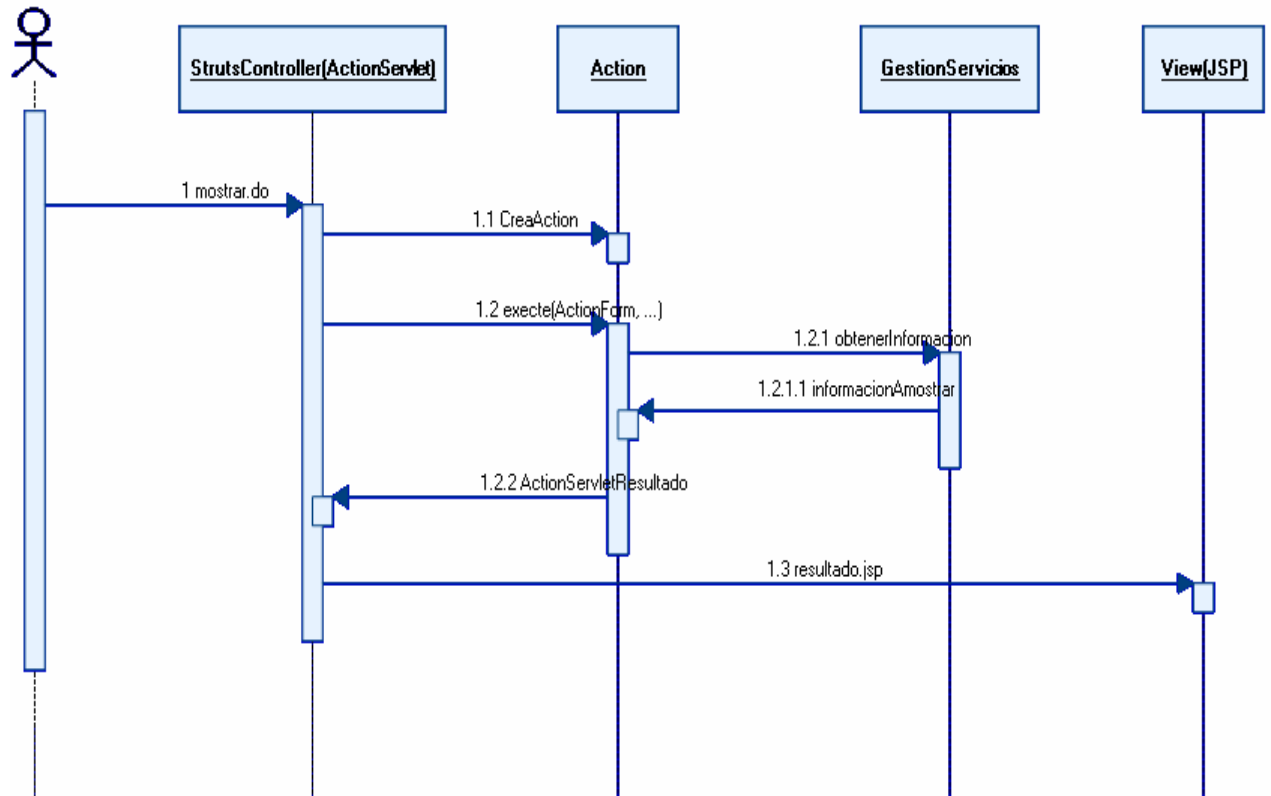


Figura 4-21: Diagrama de secuencia de consulta

Cuando un usuario realiza una operación de consulta se realiza una consulta a la Base de Datos. Las clases que se encargan de acceder a la información de la Base de Datos son las clases: *GestionServicios*, *GestionServiciosI* y *GestionServiciosI*. Por lo tanto, el *Action* puede acceder a la información mediante una de estas tres clases, tendrá que elegirla en base al idioma en el que se esté navegando en la aplicación.

A continuación, se muestra el diagrama de secuencia que se ha diseñado para la operación de añadir información de los documentos, noticia y agenda. En este caso, utilizamos los métodos definidos para subir un documento.

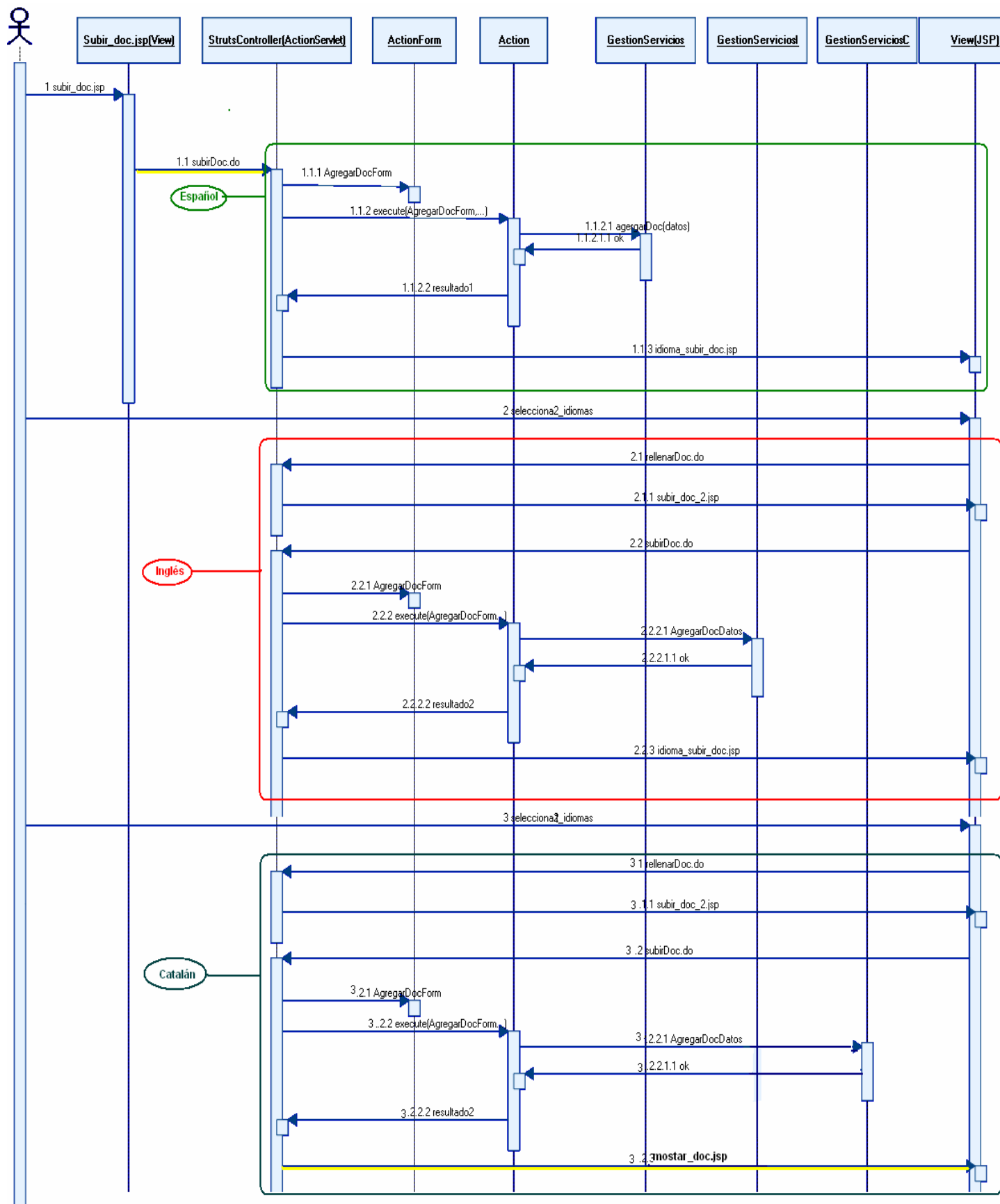


Figura 4-22: Diagrama de secuencia de añadir información (1)

Se pueden realizar cuatro operaciones de añadir elementos en la Base de Datos:

- Añadir reservas
- Añadir documentos
- Añadir información de noticias
- Añadir información de agenda

El diagrama mostrado en la *Figura 4-23: Diagrama de secuencia de añadir información (1)*, corresponde a la operación de inserta un documento, como hemos comentado, esta secuencia de operaciones es la misma que se realiza para añadir la información de noticias y de agenda. Ya que en todos ellos es necesario añadir la información en 3 idiomas.

Cuando se inserta la información en un idioma, se muestra una página que nos informa del estado de la operación realizada, mediante un mensaje, y los idiomas que quedan pendientes por añadir la información. Esta página es **idiomas_subir_doc.jsp**:

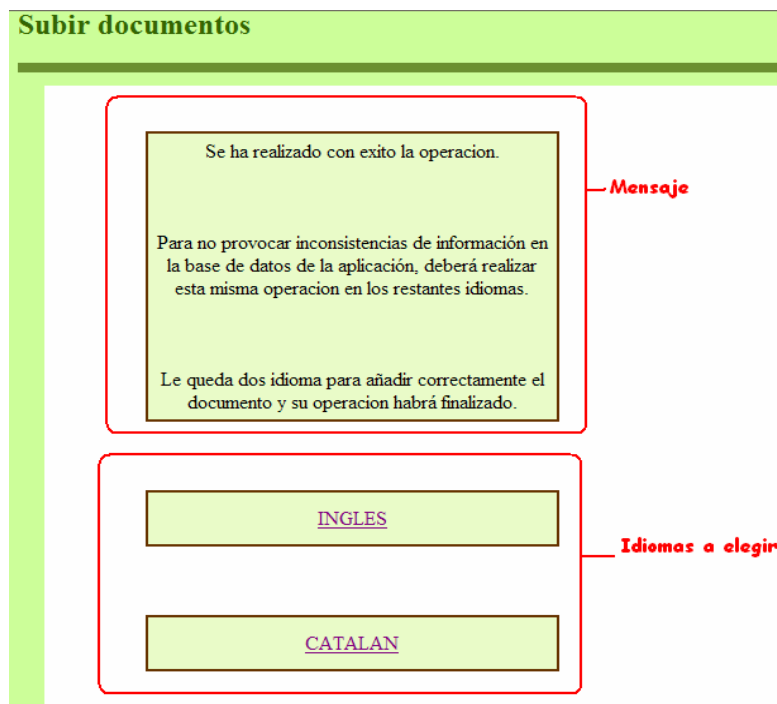


Figura 4-23: Idiomas_subir_doc.jsp

Cuando se realiza una operación de inserción en la que se obliga a insertar la información en los tres idiomas, no se muestra como una opción inicial la selección de uno de los tres idiomas ya que el primer idioma en el que se añade la información es en el que se está navegando la aplicación. Pero si mostramos la opción de seleccionar los



idiomas pendientes. Por eso, la página **idiomas_subir_doc.jsp** se muestra 2 veces (*secuencia 1.1.3 idiomas_subir_doc. Jsp* y *2.2.3 idiomas_subir_doc.jsp del diagrama de secuencia de la figura 4-25*): cuando indica que te quedan pendientes *dos* idiomas en los que insertar la información y una vez insertada la información en un segundo idioma, vuelve a mostrarse la página **idiomas_subir_doc.jsp** indicando que sólo tienes *un* idioma pendiente.

Cuando se ha finalizado la operación con éxito, es decir, que se inserta correctamente la información en los 3 idiomas, se muestra el resultado de la inserción. Para mostrar el resultado el *Action* utiliza una de estas páginas JSP's:

- **mostrar_doc.jsp**: en el caso de que la operación realizada sea añadir documento. Se muestra una lista de documentos en el que se podrá verificar que se ha insertado el mismo.
- **mostrar_age.jsp**: en el caso de que la operación realizada sea añadir información en agenda. Se muestra una lista con los contenidos de la agenda en la que se podrá verificar que se ha insertado la información.
- **mostrar_not.jsp**: en el caso de que la operación realizada sea añadir una noticia. Se muestra una lista de las noticias existentes en la que se podrá verificar que se ha insertado la noticia.

A continuación comentamos el caso de la operación añadir una reserva, ya que es una inserción distinta a las inserciones anteriores (de un documento, de un evento en la agenda o de una noticia) porque la información de las reservas sólo se almacena en la base de datos en un idioma, sólo existe una entidad de reservas, ya que la información que contiene son datos numéricos, horas y fechas y no era necesario tener tres entidades reservas para cada idioma:

En la “Figura 4-25 Diagrama de secuencia añadir reserva” se muestra el diagrama de secuencia de la operación añadir reserva de manera genérica.

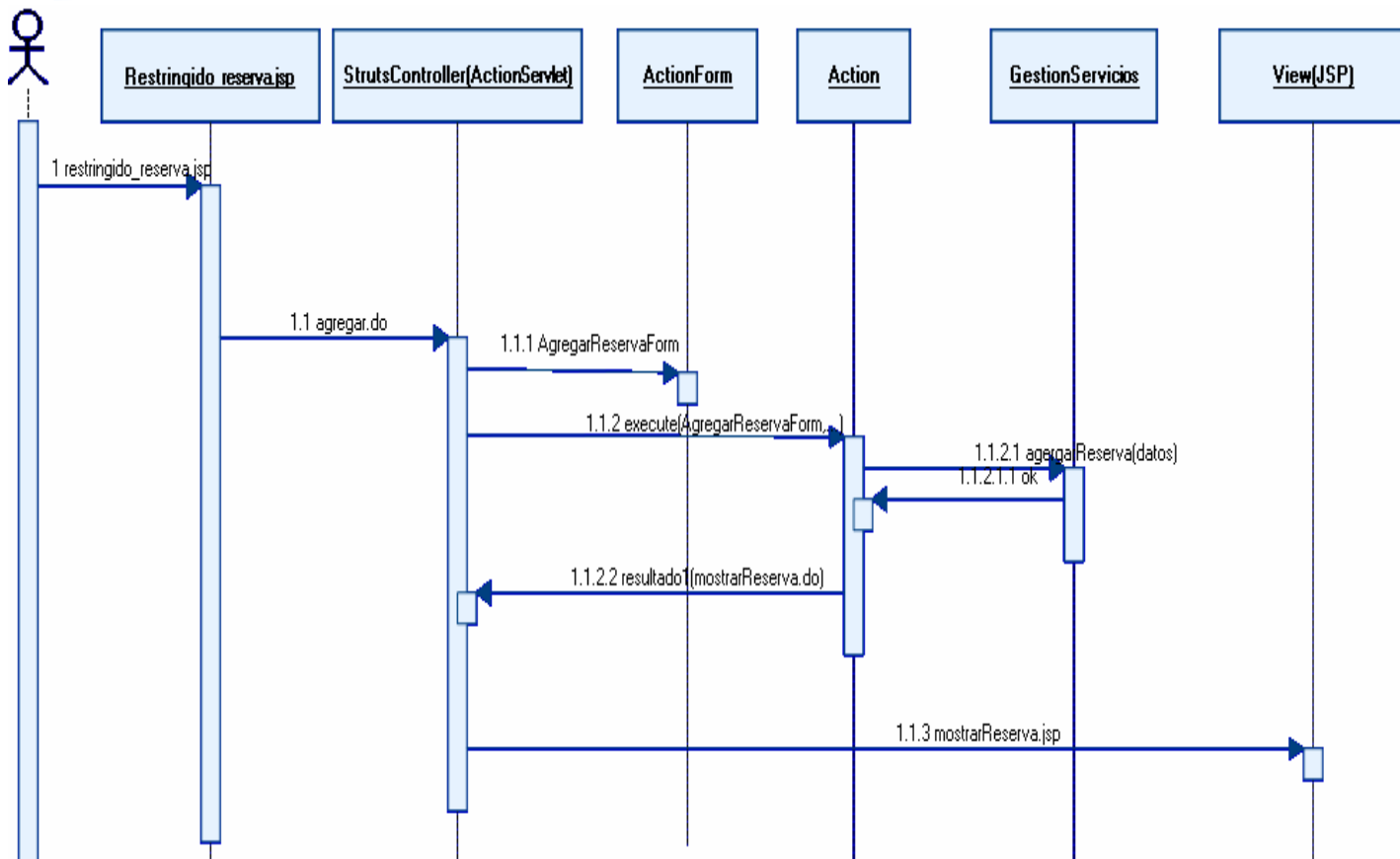


Figura 4-24: Diagrama de secuencia añadir reserva

Una vez comentados los diagramas de secuencias de las operaciones de *añadir* y *consultar* información, queda indicar cuál es la secuencia de las *operaciones de borrar y modificar*. Estas operaciones se corresponde con el diagrama de secuencia de la tecnología de Struts (*Figura 4-21: Diagrama de secuencia principal*).

La única salvedad es que en las operaciones de *modificar*, cuando se modifica un dato se modifica la información en los tres idiomas, de manera *transparente al usuario*. Esta operación se realiza en el *Action*. Y con las operaciones de *borrar* también ocurre lo mismo, cuando se borra una reserva, noticia o agenda, se borra la información en los tres idiomas (y esta operación se realiza en el *Action*).

El objetivo de borrar o modificar la información en *tres* idiomas en *una* sola operación es el de mantener la consistencia de información en la Base de Datos de manera transparente al mismo y evitar que un usuario tenga que entrar en la aplicación con el objetivo de ir actualizando información en los tres idiomas.



5. GESTIÓN DEL PROYECTO

5.1 PLAN DE TRABAJO

A continuación mostramos las fechas y los hitos marcados para el desarrollo del proyecto. Para el control y gestión del mismo se han utilizado herramientas de *google*, en este caso “Google Docs”, un documento compartido con mi directora del proyecto, sobre el cual hemos ido modificando durante el desarrollo del mismo.

Tareas	Planificado		Re-planificado		Real		Tiempo dedicado	Observaciones
	Desde	Hasta	Desde	Hasta	Desde	Hasta		
Estudio preliminar								
Estado de la cuestión	24/2/2008	1/3/2008			24/2/2008	1/3/2008	2h/día	Revisar tecnologías, métodos, especificaciones y herramientas existentes
Planteamiento del problema	2/3/2008	3/3/2008			2/3/2008	3/3/2008	3h/día	Breve descripción del problema. Habitualmente será yo quién indique el problema
Esbozo solución	4/3/2008	6/3/2008			4/3/2008	6/3/2008	3h/día	Breve descripción de la solución. Habitualmente será yo quién indique como se
Desarrollo								
Definición de requisitos	7/3/2008	24/3/2008			7/3/2008	24/3/2008	5h/día	Definición de los requisitos del sistema.
Especificación de requisitos	7/3/2008	4/4/2008			7/3/2008	4/4/2008	5h/día	Análisis de los requisitos y definición del modelo conceptual.
Diseño	7/4/2008	10/4/2008	10/4/2008	12/4/2008	7/4/2008	12/4/2008	5h/día	Diseño del sistema: arquitectura, modelo de negocio, modelo de datos, interfaz
Implementación	14/4/2008	8/6/2008	8/6/2008	28/7/2008	14/4/2008	28/11/2008	5h/día	Programación de la solución.
Pruebas	10/4/2008	8/6/2008	10/4/2008	28/7/2008	14/4/2008	30/11/2008	1h/día	Fase de pruebas.
Defensa								
Memoria	15/8/2008	30/8/2008			15/8/2008	30/8/2008	5h/día	Redacción de la memoria del proyecto.
Introducción	15/8/2008	15/8/2008			15/8/2008	15/8/2008	5h/día	Introducción al proyecto. Objetivos.
Estado de la cuestión	15/8/2008	15/8/2008			15/8/2008	15/8/2008	5h/día	Redacción del trabajo realizado en el estudio preliminar.
Planteamiento del problema	16/8/2008	16/8/2008			16/8/2008	16/8/2008	5h/día	Extensión del trabajo realizado en el estudio preliminar.
Gestión del proyecto	16/8/2008	16/8/2008			16/8/2008	16/8/2008	5h/día	Forma en la cual se ha realizado el proyecto.
Solución: Descripción / Fases del desarrollo	17/8/2008	27/8/2008	7/9/2008	3/11/2008	27/10/2008	3/11/2008	5h/día	Explicar la solución desarrollada.
Evaluación	27/08/2008	29/08/2008			27/08/2008	29/08/2008	5h/día	La evaluación dependerá del tipo de solución.
Conclusiones	29/08/2008	31/08/2008	3/11/2008	4/11/2008	3/11/2008	4/11/2008	2h/día	Conclusiones personales sobre la solución desarrollada.
Anexo	28/08/2008	1/9/2008			28/08/2008	1/9/2008	3h/día	Manual de usuario y documentos mandados por el cliente
Presentación	11/11/2008	12/11/2008			30/03/2009	3/4/2009	3h/día	Elaboración de la presentación.

Tabla 5-1: Planificación del proyecto

Se comenta las tareas contempladas en la planificación:

1. Estudio preliminar:
 - a. Estado de la cuestión: se alcanza una visión preliminar del proyecto a desarrollar.
 - b. Planteamiento del problema: mediante la petición inicial del cliente, se plantea el problema a resolver.
 - c. Esbozo de la solución: Se plantea una solución en rasgos generales.



2. Desarrollo:

- a. Definición de requisitos: se definen los requisitos de usuario, y se desarrolla un Documento de propuesta al cliente para poder extraer los requisitos. También se realizan unos prototipos de bajo nivel en Documento de propuesta que además de facilitarnos la extracción de requisitos, nos ayuda para la fase del diseño.
 - b. Especificación de requisitos: se realiza un análisis detallado de requisitos y una definición del modelo conceptual a partir de la respuesta del cliente de la Documento de propuesta.
 - c. Diseño: Se estudia las distintas maneras de poder desarrollar la aplicación, tecnologías, arquitecturas, se mejoran los prototipos a mostrar y se diseña la Base de Datos y un diagrama de clases que satisfacen nuestros requisitos.
 - d. Programación de la solución: cuando se ha definido el diseño y se sabe la arquitectura que se va a desarrollar y las tecnologías a emplear, pasamos a la fase de implementación de la solución. La mayor parte de esta fase se desarrolla desde abril hasta junio, aunque en septiembre se realizan unas modificaciones que duran hasta a finales de octubre.
 - e. Pruebas: Como se muestra en la figura la fase de pruebas ha sido desarrollada paralelamente a la fase de programación. El motivo es porque al utilizar los “Junits” para probar el modelo de datos, según se desarrollaban funcionalidades se iba verificando el correcto funcionamiento de las mismas. Y las pruebas finales que verifican el correcto funcionamiento de la aplicación, se realizaron a partir de noviembre hasta finales del mismo.
3. Defensa: en esta fase se desarrolla el documento actual y se elabora la presentación. Como se puede observar, hay una *replanificación* de la memoria, ya que las modificaciones del código obligan a actualizaciones en la memoria. El índice definido para la memoria del proyecto:
- a. Introducción
 - b. Estado de la cuestión
 - c. Planteamiento del problema



- d. Solución desarrollada
- e. Gestión del proyecto
- f. Evaluación
- g. Conclusiones
- h. Anexos

5.2 DIAGRAMA DE GANTT

Se muestra el diagrama de Gantt asociado a la planificación del trabajo.

Las tareas asociadas se corresponden con las descritas en el apartado *5.1 Plan de Trabajo*, se engloban en tres grupos:

1. Estudio Preliminar, esta definido por las subtarefas:
 - a. Estado de la cuestión
 - b. Planteamiento del problema
 - c. Esbozo solución
2. Desarrollo, esta definido por las subtarefas:
 - a. Definición de requisitos
 - b. Especificación de requisitos
 - c. Diseño
 - d. Implementación
 - e. Pruebas
3. Defensa, esta definido por las subtarefas:
 - a. Memoria
 - b. Introducción
 - c. Estado de la cuestión
 - d. Planteamiento del problema
 - e. Gestión del proyecto
 - f. Solución
 - g. Evaluación
 - h. Conclusiones
 - i. Anexos



	i	Nombre de tarea	Duración	Comienzo	Fin	Predece:
1		<input type="checkbox"/> ESTUDIO PRELIMINAR	12 días	dom 24/02/08	jue 06/03/08	
2		Estado de la cuestión	7 días	dom 24/02/08	sáb 01/03/08	
3		Planteamiento del problema	2 días	dom 02/03/08	lun 03/03/08	2
4		Esbozo solución	3 días	mar 04/03/08	jue 06/03/08	3
5		<input type="checkbox"/> DESARROLLO	197 días?	vie 07/03/08	dom 30/11/08	4
6		Definición de requisitos	12 días	vie 07/03/08	lun 24/03/08	4
7		Especificación de requisitos	21 días	vie 07/03/08	vie 04/04/08	
8		Diseño	6 días	lun 07/04/08	sáb 12/04/08	7
9		Implementación	168 días	lun 14/04/08	vie 28/11/08	8
10		Pruebas	169 días?	lun 14/04/08	dom 30/11/08	
11		<input type="checkbox"/> DEFENSA	61 días	vie 15/08/08	mar 04/11/08	
12		Memoria	13 días	vie 15/08/08	sáb 30/08/08	
13		Introducción	1 día	vie 15/08/08	vie 15/08/08	
14		Estado de la cuestión	1 día	vie 15/08/08	vie 15/08/08	
15		Planteamiento del problema	1 día	sáb 16/08/08	sáb 16/08/08	14
16		Gestión del proyecto	1 día	sáb 16/08/08	sáb 16/08/08	13
17		Solución: Descripción / Fases	6 días	lun 27/10/08	lun 03/11/08	16
18		Evaluación	3 días	mié 27/08/08	vie 29/08/08	
19		Conclusiones	2 días	lun 03/11/08	mar 04/11/08	18
20		Anexo	4 días	jue 28/08/08	lun 01/09/08	

Tabla 5-2: Tareas Gantt

La siguiente figura muestra la gráfica asociada a las tareas:

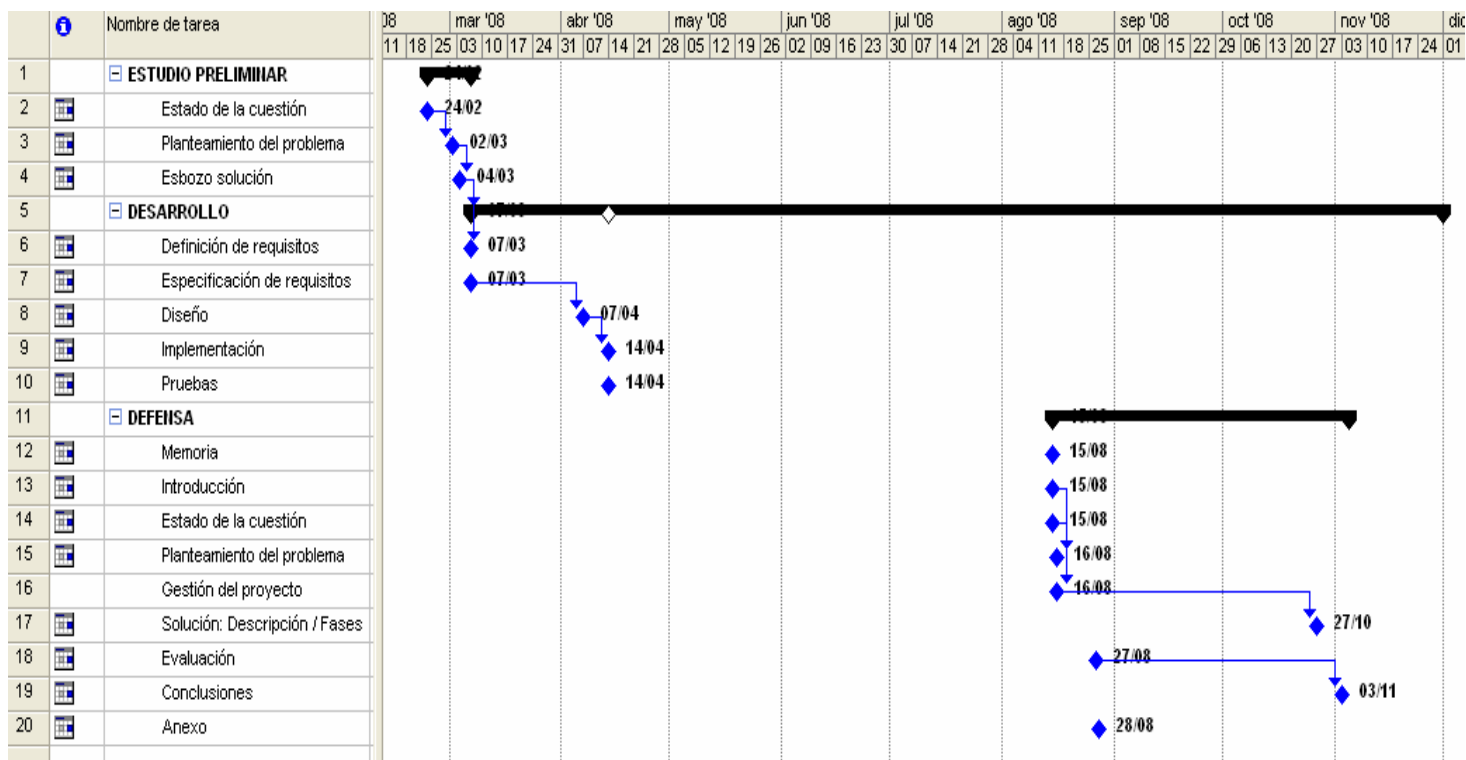


Tabla 5-3: Diagrama de Gantt-Hitos



5.3 ANÁLISIS ECONÓMICO

Para estimar el coste de la aplicación desarrollada se han utilizado los criterios que influyen en cualquier proyecto de consultoría, como es el tiempo de desarrollo, los costes derivados del consumo de recursos y la infraestructura utilizada para el desarrollo. También hemos de tener en cuenta aspectos como la formación necesaria y mantenimiento de la aplicación.

El resumen del presupuesto que vamos a mostrar a continuación, es un **presupuesto ficticio** ya que estima que el proyecto va a ser realizado por analistas, jefes de proyecto, etc. Pero en realidad, al ser un proyecto académico ha sido realizado sólo por la autora del presente documento. La principal diferencia con el presupuesto que mostraremos y el real es el coste de las horas realizadas por los empleados, más adelante se comentará este aspecto.

Mostramos la estimación de costes de la aplicación teniendo en cuenta los costes directos de personal, costes directos materiales y los costes indirectos del proyecto.

5.3.1 DESGLOSE POR ACTIVIDADES DEL PROYECTO

Se han puesto las horas totales de cada una de las fases del proyecto teniendo en cuenta el tiempo empleado en su desarrollo. Esta tabla se basa en la “Figura 5-1: Planificación del proyecto” y en la “Figura 5-2: Tareas Gantt”.

DURACIÓN PROYECTO	
Actividades	Tiempo real
Planificación	14h (Estado de la cuestión = 7días * 2h/día)
Estudio Viabilidad Sistema	15h (Entrenamiento del problema+Esbozo solución = 5 días * 3h/día)
Análisis y diseño	135h (definición+especificación de los requisitos + diseño = 27días * 5h/día)



Implementación	840h (Implementación=168días*5h/día)
Documentación	141h (13días *5h/días + 2días *2h/días + 4*3h/día)
Pruebas	169h (Pruebas= 169días*1h/día)
Implantación	40h (Estimación)
Total horas/ empleado	1.254h

Tabla 5-4. Estimación Tiempo dedicado a las distintas actividades del Proyecto

5.3.2 SALARIOS POR CATEGORÍA

Para conocer el gasto de la empresa en relación al trabajo de sus empleados o personal que conforma la misma es necesario conocer los salarios de los trabajadores. Para saber los salarios hemos tomando como referencia una empresa que se dedica a la elaboración de portales Web.

Los salarios medios se definen en la siguiente tabla:

Coste salarios de personal			
Cargo	Sueldo Bruto Año	Sueldo Bruto mes	Coste Hora/empleado
Jefe de Proyecto	42.000€	3.500 €	21,875 €
Analista	30.000 €	2.500 €	15,625 €
Programador	22.000 €	1.833,33 €	11,45 €
Gestor de pruebas	22.000 €	1.833,33 €	11,45 €

Tabla 5-5. Coste Salarios puesto/hora

Mostramos otra tabla con los porcentajes que van a influir en el coste total del proyecto. Esta tabla se toma como referencia de la misma empresa de la que proviene la tabla de los costes de los salarios puesto/hora.



ACTIVIDAD	PORCENTAJE
<i>Costes directos</i>	-
<i>Costes Indirectos</i>	10 %
<i>Riesgo</i>	15 %
<i>Beneficios</i>	40 %
<i>IVA</i>	16 %

Tabla 5-6. Porcentajes

A continuación, una vez conocidos los salarios, estimamos la dedicación de los empleados en cada una de las fases definidas en el proyecto. De esta manera obtendremos el coste del personal por cada fase.

Coste salarios de personal/Fase					
Cargo	Jefe de Proyecto	Analista	Programador	Gestor de pruebas	TOTAL
<i>Planificación</i>	14h				306,25€ (14*21,875)
<i>Estudio Viabilidad Sistema</i>	15 h				328,125 €
<i>Análisis y diseño</i>	67,5h (135h/2)	67,5h (135h/2)			2.531,25 €
<i>Implementación</i>		223,6 (671h/3)	447,3 (671h-671h/3)		8.615,335 €
<i>Documentación</i>	47h (141h/3)	94h (141h-141h/3)			2.496,875€
<i>Pruebas</i>				169h	1.935,05€
<i>Implantación</i>		40 h			625 €
TOTAL	143,5*21,875=3 .139,06€	425,1*15,625=6 .642,1875€	447,3*11,45=5.121, 585€	169*11,45=1.93 5,05€	16.837,885 €

Tabla 5-7. Coste salarios de personal/fase



5.3.3 GASTOS DE PERSONAL IMPUTABLES AL PROYECTO

A partir de la tabla anterior es posible sacar la relación entre costes de los empleados del proyecto y las horas que van a dedicar:

GASTOS HORAS/EMPLEADOS		
Empleado	Horas	Coste
<i>Jefe de Proyecto</i>	143h	$143,5 \times 21,875 = 3.139,06\text{€}$
<i>Analista</i>	425,1 h	$425,1 \times 15,625 = 6.642,1875\text{€}$
<i>Programador</i>	447,3h	$447,3 \times 11,45 = 5.121,585\text{€}$
<i>Gestor de pruebas</i>	169h	$169 \times 11,45 = 1.935,05\text{€}$
TOTAL	1.184,9h	16.837,885 €

Tabla 5-8. Coste horas/empleados

5.3.4 GASTOS DIRECTOS MATERIALES

Para la realización de este proyecto se han empleado los siguientes medios materiales:

GASTOS MATERIALES		
Material	Cantidad	Coste
<i>Ordenador Portátil Dell Inspiron 64 bits 1 MB RAM 60 GB Disco Duro</i>	1	499 €
<i>Cable RJ 45 (metro)</i>	1	1 €
<i>Sistema Operativo Windows XP Proffesional</i>	1	505 €
TOTAL	-	1.005 €

Tabla 5-9. Gasto Materiales



Además de estos materiales mostrados en la tabla necesitamos otras herramientas que son gratuitas por eso no las mencionamos anteriormente:

- Servidor Web y BBDD: Se requiere como servidor Web Apache y se utiliza como BBDD MySQL, ambos son de libre distribución (por lo que el coste asociado es 0).
- Licencias software: el framework de programación (Struts), el lenguaje de programación utilizado (Java), así como el motor de persistencia utilizado (Hibernate), son también de libre distribución (por lo que el coste asociado es 0).

5.3.5 GASTOS INDIRECTOS

Los gastos indirectos suponen el 10 % de los gastos totales en el proyecto, ya que tomamos como referencia los porcentajes que se indican en la “Tabla3.Porcentajes”. En la siguiente tabla se especifican todos los componentes de los gastos indirectos y una estimación de los posibles costes:

Gastos Indirectos	
Gasto	Coste
<i>Limpieza de la oficina</i>	100€ (Incluido en los costes indirectos)
<i>Electricidad</i>	200€(Incluido en los costes indirectos)
<i>Agua</i>	100€(Incluido en los costes indirectos)
<i>Amortizaciones</i>	100€ (Incluido en los costes indirectos)
<i>Alquiler Local</i>	700€(Incluido en los costes indirectos)
<i>Telefono y ADSL</i>	200€(Incluido en los costes indirectos)

Tabla 5-10. Gastos Indirectos



5.3.6 GASTOS DIRECTOS

En esta tabla que se muestra a continuación se especifican todos los gastos obtenidos anteriormente para la realización del resumen del presupuesto

Gastos Directos	
Gasto	Coste
<i>Empleados</i>	16.837,885 €
<i>Gastos en materiales</i>	1.005 €
<i>TOTAL</i>	17.842,885 €

Tabla 5-11. Gastos Directos



5.3.7 RESUMEN DEL PRESUPUESTO

El presupuesto final se elabora teniendo en cuenta el presupuesto del proyecto (gastos directos) junto con el beneficio de la empresa 40%, los gastos indirectos (electricidad, adsl, utilización de local...) y unos posibles riesgos del 15%. Al precio final se le debe añadir el 16% de IVA.

Presupuesto			
ACTIVIDAD	PORCENTAJE	SUBTOTAL	TOTAL
<i>Costes directos</i>	-	17.842,885 €	17.842,885 €
<i>Costes Indirectos</i>	10 %	1.784,288 € (10% de 17.842,885)	19.627,173€
<i>Riesgo</i>	15 %	2.944,07€ (15% de 19.627,1735)	22.571,246€
<i>Beneficios</i>	40 %	9.028,498€ (40% de 22.571,246)	31.599,74€
<i>IVA</i>	16 %	5.056€ (16% de 31.599,74)	36.655,7€
Total			36.655,7 €

Tabla 5-12. Presupuesto

El presupuesto final asciende a un TOTAL de **36.655,7 €IVA INC.**



5.3.8 RESUMEN DEL PRESUPUESTO REAL

El presupuesto que mostramos a continuación consiste en considerar que todas las tareas que interviene en la elaboración del proyecto sólo las realiza una persona, que es la autora del proyecto y el coste por hora será el de un programador. Por lo tanto obtendremos un presupuesto más económico, ya que el coste por hora del programador es más bajo que el del analista o el del jefe de proyecto.

5.3.8.1 ESFUERZO DEL PROYECTO

Se ha realizado una estimación de las líneas de código codificadas en el desarrollo del PFC en sus tres partes fundamentales de desarrollo software:

- Vistas (archivos JSP)
- Modelo (archivos JAVA)
- Controlador (archivo XML + archivos Action y Forms)

Para realizar la estimación se ha calculado el tamaño total de los archivos codificados y se ha calculado realizando una media tras coger varios archivos de ejemplo que cada KByte en disco se trata de veinticinco líneas de código.

Se tiene un tamaño total de desarrollo de 652 KBytes de ficheros de texto plano. Esto implica que se ha realizado una codificación de **16300 líneas** escritas en los siguientes lenguajes: JSP y Java.

5.3.8.2 COSTE EFECTIVO DEL PROYECTO

El resumen del presupuesto mostrado anteriormente es un presupuesto ficticio ya que estima que el proyecto va a ser realizado por analistas, jefes de proyecto, etc. cuando en realidad el proyecto ha sido realizado por la autora del presente documento. La principal diferencia con el presupuesto anterior es el coste de las horas realizadas por los empleados. En realidad el coste sería el siguiente:

COSTE HORAS/EMPLEADOS		
Empleado	Horas	Coste
Nieves Pemán Baena	1.184,9h	1.184,9h * 11,45=13.155€

Tabla 5-13: Coste empleado/hora



Donde las horas serían las correspondientes a las realizadas en un proyecto por los analistas, jefes de proyecto, etc. Pero considerando el coste por hora como si fuera un programador.

El presupuesto final REAL es el siguiente:

Presupuesto			
ACTIVIDAD	PORCENTAJE	SUBTOTAL	TOTAL
<i>Costes directos</i>	-	14.160€ (1.005 gastos directos material + 13.155 coste empleados)	14.160€
<i>Costes Indirectos</i>	10 %	1.416€ (10% de 14.160)	15.576€
<i>Riesgo</i>	15 %	2.336,46€ (15% de 15.576)	17.912,4€
<i>Beneficios</i>	40 %	7.164€ (40% de 17.912,4)	25.077€
<i>IVA</i>	16 %	4.012,37€ (16% de 25.077)	29.089,3€
Total			29.089,3€

Tabla 5-14. Presupuesto Real

El presupuesto final real asciende a un TOTAL de **29.089,3 €IVA INC.**



6. CONCLUSIONES

Se ha realizado un gestor de contenidos que cubre las necesidades que nos planteaba el cliente en su petición inicial, un portal Web para la gestión la información que el departamento quiere mostrar a cualquier usuario y la gestión de información que sólo es para el personal del departamento. A demás ofrece a los miembros del departamento:

- Gestionar reservas (de cañones, laboratorios, coches y ordenadores).
- Gestionar la información de noticias y agenda tanto las públicas como las privadas. Considerándose información pública aquella que se muestra en la página principal del portal Web y la que cualquier usuario sin registrarse puede observar; mientras que la información privada es la que sólo los miembros del departamento pueden ver una vez acceden a las páginas de usuarios registrados en el portal.
- Gestionar documentos propios del departamento

6.1 CONCLUSIONES

Varios son los beneficios que se obtiene de esta herramienta, cubriendo diferentes aspectos de negocio:

- Se satisfacen las necesidades del cliente puesto que la aplicación incluye las funcionalidades de consulta y de gestión.
- Incrementar la facilidad de la gestión de reservas, de la publicación de documentos y publicación de información de noticias y agenda. Ya que se utilizan elementos en los formularios que hace más sencilla la inserción o modificación de información, como son los campos desplegables y además se muestra una *opción de ayuda* en la que se define los campos del formulario.



- Simplifica y optimiza los procesos de modificación y borrado de datos. En el caso de las operaciones de modificación o borrado que se actualiza la información en los tres idiomas de manera transparente al usuario.
- El usuario puede estar informado del estado de las operaciones que realiza, por ejemplo cuando inserta información o borra.
- La posibilidad de consultar cualquier duda a la hora de realizar modificaciones o inserciones de datos. Esto se satisface gracias a las *opciones de ayuda*.
- Se consigue presentar la información de manera estructurada, las operaciones tienen unos nombres muy descriptivos para que sean más fáciles de intuir por el usuario y sino tiene una opción de ayuda.
- Se dispone de tres idiomas para mostrar la presentación de la información.
- No se requiere una figura de “usuario traductor” para la traducción de información en los distintos idiomas. El propio usuario lo puede realizar en una operación de insertar información. Para ello se le facilita un formulario con la información “casi completa”, así sólo tiene que añadir “títulos” y/o “descripciones”, contenidos que si varían en función de un idioma u otro. Este hecho es una ventaja si el usuario es políglota, en nuestro caso, la mayoría de los usuarios del departamento lo son.

De la aplicación obtenida se abre un conjunto de posibles líneas de ampliación. Entre ellas citamos las que nos parecen de mayor interés:

6.2 TRABAJOS FUTUROS

- La aplicación podría tener una opción que fuera crear currículos de los profesores.
- Se podrían crear unas funcionalidades para que el usuario administrador, una vez que entra en el portal y se registra, pudiera tener



funcionalidades para insertar información que es fija en el portal como pueden ser las titulaciones que imparten, las asignaturas que se cursan, los masters, las tesis, etc.

- Cuando se muestra los profesores de contacto en los proyectos o trabajos dirigidos, también se indica el e-mail de los mismos, pues se podría crear una opción de gestor de e-mail para que cuando se selecciona con el ratón el e-mail del profesor, se pudiera redactar uno en ese momento.
- Para que la traducción de la información añadida en la aplicación fuera totalmente independiente del usuario que la inserta, se podría crear una figura llamada “usuario traductor” a la que le llegase un e-mail indicándole que se ha añadido información, de esta manera, éste se conectaría a la aplicación y se encargaría de traducir el contenido en los distintos idiomas.
- En la operación “cambiar contraseña de usuario”, se podría hacer una verificación, como realizan algunas aplicaciones con una palabra secreta o película favorita, en la que se comprueba si la persona que quiere cambiar la contraseña de acceso es la propietaria de la cuenta. Una vez verificado se le enviaría esta nueva contraseña en un correo electrónico al usuario.
- La funcionalidad que ofrece la opción “[+] información” en las secciones de noticias y agenda hace referencia a “información” adicional que se encuentra en una página Web, se podría contemplar la opción de almacenar esta información adicional en ficheros de texto. Esta posible mejora ya se ofrece en la aplicación pero sólo para la funcionalidad “subir documentos”, que pueden tener una referencia a una página Web o bien puede ser un fichero de texto que se debe guardar en la aplicación.



6.3 CONCLUSIONES PERSONALES

Cuando comencé el proyecto pensé que iba a tardar menos tiempo de lo que he tardado en terminarlo, ya que surgieron dificultades en la fase de implementación, especialmente con la tecnología de *Hibernate* además no se contemplaron ciertas funcionalidades al principio del proyecto que más tarde se decidieron implementar.

Los problemas que tuve con *Hibernate* se deben a que utilice las opciones que ofrece el editor *Eclipse* para generar el modelo de datos mediante el motor de persistencia *Hibernate* y debían de faltar “pluggins”, ya que fallaba su generación. Pero finalmente se solucionó y se pudo empezar a desarrollar sin problemas. Considero que he cumplido mis objetivos personales ya que:

- Se han aplicado los conocimientos adquiridos en la titulación, especialmente los referidos al desarrollo, la ingeniería del software, la programación orientada a objetos y los lenguajes de modelado.
- He profundizado y mejorado las habilidades de trabajo en los entornos de desarrollo, los patrones de diseño y la arquitectura Modelo-Vista-Controlador. Y he ampliado los conocimientos de tecnologías Struts e *Hibernate*.
- Se ha conseguido realizar un proyecto de acuerdo a las especificaciones definidas que finalmente será implantado.
- He ampliado mis conocimientos de las funcionalidades que nos ofrecen los editores de páginas Web, como es el caso del *Dreamweaver*, y poder realizar diseños más elaborados.

Por lo que considero que he aprendido mucho haciendo el proyecto fin de carrera aunque ha sido duro tener que compatibilizarlo con la vida laboral.

También he de añadir que la tutorización de mi directora de proyecto ha sido de gran ayuda para la elaboración del mismo ya que su experiencia, su exigencia, su dedicación y su metodología de trabajo siempre han ido orientada a elaborar el proyecto, en todas las fases de su desarrollo, con la mayor calidad posible.



7. BIBLIOGRAFÍA

- [1] Antonio J. Martín Sierra: *Struts*, 1ª Edición, Editorial RAMA, ISBN: 978-84-7897-855
- [2] Chuck Cavaness: *Jakarta Struts.*, 2ª Edición, Editorial O'Reilly, ISBN 0596006519
- [3] Damon Hougland; Aaron Tavistock: *Core JSP*, Editorial Prentice Hall, ISBN-10: 0-13-088248-8
- [4] David Camacho, Jose M^a Valls: *Programación, Algoritmos y ejercicios resueltos en Java*, 1ª Edición, Editorial Anaya, ISBN: 9788420540245
- [5] Eric y Gradeckil, Joseph D. y Hightower, Richard: *Profesional Hibernate*, 1ª Edición, ISBN: 978076457677
- [6] James Holmes: *Struts: The Complete Reference*, 2ª Edition, Editorial McGraw Hill, ISBN: 9780072262407
- [7] Mukhar, Kevin y Lauinger, Todd y Carnell, Jonh : *Fundamentos de Bases de Datos con JAVA: JDBC, SQL, J2EE, EJB, JSP , XML*, 1ª Edición, Editorial Anaya, ISBN: 9788441513624
- [8] Chistopher, Schmitt: *Curso de CSS*, 1ª Edición, Editorial O REILLY, ISBN: 9788441521988
- [9] Título: Introducción a Hibernate:
http://www.froses.com/Assets/Files/Articles/Hibernate_Introduccion_es.pdf
- [10] Título: JDO:
<http://dis.um.es/~jmolina/Persistencia%20de%20Objeto%20JDO.pdf>
- [11] Título: Introducción a los Sistemas de Gestión de Contenidos (CMS) de código abierto.
<http://mosaic.uoc.edu/articulos/cms1204.html>
- [12] Título: Escribir documentos en HTML.



<http://www.uv.es/jac/guia/notas.htm>

[13] Título: HTML

<http://www.lcc.uma.es/~eat/services/html-js/manual14.html>

[14] Título: Definición XML

<http://www.masadelante.com/faq-xml.htm>

[15] Título: ¿Qué es JSP?

<http://www.desarrolloweb.com/articulos/831.php>



8. DEFINICIONES Y ACRÓNIMOS

A lo largo del documento se usan una serie de términos que pueden no contar con el conocimiento del cliente; de ahí, la necesidad de aclararlos:

8.1 ACRÓNIMOS

Acrónimo	Descripción
UML	Unified Modeling Language
JDO	Java Data Objects
HTML	HyperText Markup Language
JDBC	Java DataBase Connectivity
JSP	Java Server Pages
J2EE	Java 2 Platform, Enterprise Edition
MVC	Modelo Vista Controlador
XML	eXtensible Markup Language

Tabla 8-1: Acrónimos

8.2 DEFINICIONES

- **Apache:** es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.
- **Commit:** transacción del lenguaje de consulta de datos SQL (*Structured query language*) que confirma los cambios realizados en la base de datos.
- **Dreamweaver:** programa que combina un conjunto de herramientas de diseño visual con funciones de edición basadas en texto HTML, para gestión, creación y mantenimiento de sitiosWeb.
- **Gestor de contenidos:** CMS (*Content Management System*) - Sistema que facilita la gestión de contenidos en todos sus aspectos: creación,



mantenimiento, publicación y presentación. También se conoce como *Web Content Management* (WCM) sistema de gestión de contenido de Web.

- **HIBERNATE:** es un motor de persistencia que permite realizar un modelo de clases partiendo de un único modelo de datos. Te permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. Es *open source*, lo que supone, entre otras cosas, que no tenemos que pagar nada por adquirirlo.
- **HTML:** (Hyper Text Markup Language) lenguaje basado en marcas que indican las características del texto, utilizado para definir documentos de hipertexto en Web., es decir, documentos de texto presentado de forma estructurada, con enlaces (links) que conducen a otros documentos o a otras fuentes de información (por ejemplo bases de datos) que pueden estar en tu propia máquina o en máquinas remotas de la red. Todo ello se puede presentar acompañado de cuantos gráficos estáticos o animados y sonidos seamos capaces de imaginar [11].
- **MySQL:** es una base de datos relacional multiplataforma de código abierto, muy utilizada en aplicaciones Web [11].
- **JDBC** (Conectividad Java de bases de datos): es un API (Application Programming Interface) para conectar a las bases de datos desde el entorno Java. Es una alternativa a ODBC ya que la interfaz de Java es más cómoda para los programadores Java que ODBC del lenguaje C interfaz [11].
- **JSP:** JavaServer Pages (JSP) es una tecnología Java que permite generar contenido dinámico para Web, en forma de documentos HTML, XML o de otro tipo. Esta tecnología es un desarrollo de la compañía Sun Microsystems. La Especificación JSP 1.2 fue la primera que se liberó y en la actualidad está disponible la Especificación JSP 2.1. Las JSP's permiten la utilización de código Java mediante scripts. Además es posible utilizar algunas acciones JSP predefinidas mediante etiquetas. Estas etiquetas pueden ser enriquecidas mediante la utilización de Librerías de Etiquetas (TagLibs o Tag Libraries) externas e incluso personalizadas [15].
- **J2EE:** Java Platform, Enterprise Edition o Java EE (anteriormente conocido como Java 2 Platform, Enterprise Edition o J2EE hasta la versión 1.4), es una plataforma de programación—parte de la Plataforma Java—para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de N niveles distribuida, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de



aplicaciones. La plataforma Java EE está definida por una especificación. Similar a otras especificaciones del Java Community Process, Java EE es también considerada informalmente como un estándar debido a que los suministradores deben cumplir ciertos requisitos de conformidad para declarar que sus productos son conformes a Java EE; estandarizado por The Java Community Process / JCP [16].

- **MVC:** Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones Web, donde la vista se representa con páginas HTML o JSP; el modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio; y el controlador es el responsable de recibir los eventos de entrada desde la vista que se puede representar con la tecnología Struts.
- **Servidor de Aplicaciones:** servidor Web que permite interaccionar al cliente con la aplicación.
- **STRUTS:** Struts es un framework de desarrollo de aplicaciones Web bajo el patrón MVC bajo la plataforma J2EE.
- **XML:** son las siglas de Extensible Markup Language, una especificación/lenguaje de programación desarrollada por el W3C. XML es una versión de SGML, diseñado especialmente para los documentos de la Web. Permite que los diseñadores creen sus propias etiquetas, permitiendo la definición, transmisión, validación e interpretación de datos entre aplicaciones y entre organizaciones [14].
- **Usuario registrado:** usuario que esta dado de alta en la aplicación.
- **Miembro:** usuario que esta dado de alta en la aplicación y pertenece al departamento.
- **Sitio público:** parte pública de la aplicación, puede acceder cualquier usuario
- **Sitio privado:** parte privada de la aplicación, se necesita estar registrado para poder acceder.



9. ANEXOS



A1 ANEXO MANUAL DE USUARIO

Este anexo corresponde con el manual de usuario que se le entrega al cliente del portal Web desarrollado para la Gestión de Información de un departamento de Investigación.



A1.1 INTRODUCCIÓN

A1.1.1 LECTORES A QUIENES VA DIRIGIDO

El manual de usuario expone los procesos que el usuario puede realizar con el sistema implantado. Para lograr esto, se detallan las formas de acceder e introducir información.

Este documento va dirigido a todos los usuarios que pretendan utilizar la aplicación, independientemente de que sean usuarios expertos o no.

A1.1.2 PROPÓSITO

El *Manual de Usuario* tiene como propósito mostrar cómo debe instalarse la aplicación y qué funcionalidades ofrece, indicando los procedimientos detallados para realizar ambas acciones.

Recordemos cual es el propósito de la aplicación o los objetivos que se pretenden conseguir, los cuales se dividen en dos grupos:

1) Funcionamiento del sistema:

Las funciones básicas del sistema son dos: consultas, inserciones, modificación y borrados de datos. A la vez hacen distinción de sitios, es decir, hay distintas funcionalidades en base a si es un usuario *público* o *registrado*, definiendo como *registrado* a todos los miembros del departamento. Las tareas que nos solicitan son las siguientes:

1. Consultar información: disponible tanto usuario registrado en la aplicación o no.
 - a. Información del personal del Departamento: PDI, PAS, Becarios.
 - b. Información de la docencia que imparte el departamento: Titulaciones, Masters y cursos.
 - c. Información de Ofertas que ofrece: Proyectos Finales de Carrera (PFC's), Becas, Trabajos dirigidos.



- d. Información de Investigación: grupos de investigación, Tesis y transferencia tecnológica.
 - e. Documentos: actas, consejos.
 - f. Información de Reservas: coches, ordenadores, laboratorios y cañones.
 - g. Información de Noticias y agenda.
 - h. Validar usuario del departamento. Lo pueden realizar sólo los *usuarios registrado* en la aplicación ya que al usuario que no esté registrado le dará un error de validación.
2. Añadir, borrar o modificar información en la aplicación, sólo la pueden realizar los *usuarios registrados*:
- a. Reservas: de coche, cañón, laboratorio y ordenador.
 - i. Crear reservas.
 - ii. Modificar reservas.
 - iii. Borrar reservas.
 - b. Documentos del Departamento: actas o consejos.
 - i. Añadir.
 - ii. Consultar.
 - c. Usuarios.
 - i. Alta (Sólo las pueden hacer los usuarios administradores).
 - ii. Bajas (Sólo las pueden hacer los usuarios administradores).
 - iii. Modificaciones.
 - d. Información de Noticias y de Agenda a la aplicación.
 - i. Añadir
 - ii. Borrar
 - iii. Modificar
3. Cerrar la sesión

2) Visualización de aplicación

La pantalla se dividirá en 3 zonas: cabecera, menú, información (que siempre esta en la parte central). En la siguiente imagen se puede observar cual será la organización de cada zona.



Figura A1- 1: Página principal

- ❖ En la **zona del menú** se mostrarán todas las operaciones que puede realizar cualquier usuario, sin necesidad de registrarse.
- ❖ En la **zona central** es donde se mostrará toda la información, según se vayan seleccionando opciones del menú.
- ❖ Y en la zona **superior**, la página principal de la aplicación posibilita cambiar de idioma: ingles, catalán o español. Además la cabecera tiene 2 enlaces, uno a la Universitat de Lleida y el otro a la página del departamento de Producció Vegetal i Ciència Forestal



Figura A1- 2: Logo cabecera



En la **zona de identificación** es donde el usuario deberá de introducir sus datos para acceder a información privada:

Figura A1- 3: Zona de identificación

A1.2 ACCESO A LA APLICACIÓN

Se accede a la aplicación introduciendo la siguiente URL en la barra de direcciones del navegador Web.

http://<nombre_maquina>:<puerto >/PFC-1

En este caso: http://localhost: 8080/PFC-1

Accedemos a la aplicación y podemos seleccionar le idioma en el que queremos operar



Universitat de Lleida
Escola Tècnica Superior d'Enginyeria Agrària
Departament de Producció Vegetal i Ciència Forestal

Language selection: [ES] [EN] [CA]

- ¿Quienes somos?
- Personal
- Docencia
- Ofertas
- Investigacion

Usuario:
Clave:
OK

¿Olvidaste tu contraseña?

Agenda

Dia: 01-02-2009 Hora:09:00 Segundo desayuno coloquio sobre conciliacion laboral y familiar [+] Información	Dia: 01-02-2009 Hora:09:00 Segundo desayuno coloquio sobre conciliacion laboral y familiar Información: http://www.uc3m.es
Dia: 25-10-2008 Hora:20:00 Seminario de economia de la empresa. Francisco Javier Husillos [+] Información	Dia: 25-10-2008 Hora:10:00 Seminario de economia de la empresa. Francisco Javier Husillos [+] Información

Noticies

1_ Seminario de economia aplicada y econometria. "Recientes resultados en Dia: 20-10-2008 Descripción: El catedrático Roberto Berino [+] Información	2_ Seminario de economia aplicada y econometria. "Recientes resultados en ... Dia: 25-10-2008 Descripción: "Se conoceran los fundamentos básicos.... [+] Información
3_ Seminario de economia aplicada y econometria. "Recientes resultados en ... Dia: 25-10-2008 Descripción: "Se mostraran los casos prácticos de seminario 2... [+] Información	4_ Seminario de economia aplicada y econometria. "Recientes resultados en Dia: 25-10-2008 Descripción: Se conoceran los fundamentos básicos... [+] Información

Figura A1- 4: Pantalla principal de la aplicación

A1.3 NAVEGAR SIN REGISTRARSE

Vamos a detallar las funcionalidades que puede realizar un usuario sin estar registrado en la aplicación.

A1.3.1 CAMBIAR DE IDIOMA

Si el usuario desea elegir el idioma de la aplicación deberá seleccionarlo en ésta página principal. Las opciones que se dan son: catalán, inglés y español. En el momento que se elija esta opción, se mantendrá a lo largo de la aplicación hasta que el usuario decida volver a cambiarlo, en ese caso, se volverá a la página principal de la aplicación con el idioma indicado.

A1.3.2 VISUALIZAR INFORMACIÓN DE LA AGENDA Y DE LAS NOTICIAS DEL DEPARTAMENTO

Para ver esta información lo único que se necesita es estar en la página principal de la aplicación. En ella se encuentra las noticias y los eventos de la agenda con fecha a partir de la actual. La información que se puede ver en cada una de ellas es:

- Cada evento de la sección de agenda consta de las siguientes partes: fecha, hora, título y dirección Web asociada con más información:

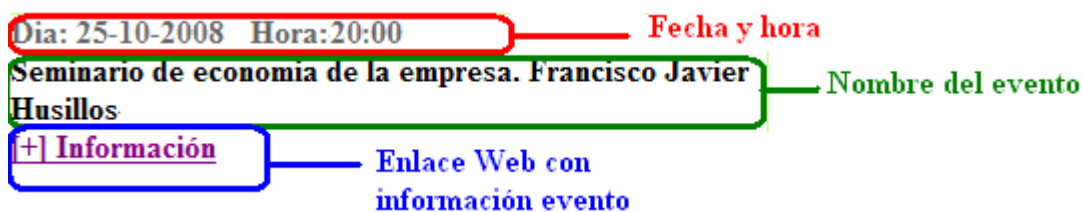


Figura A1- 5: Evento

- Cada noticia de la sección de noticias consta de las siguientes partes: fecha, título, descripción y dirección Web asociada con más información

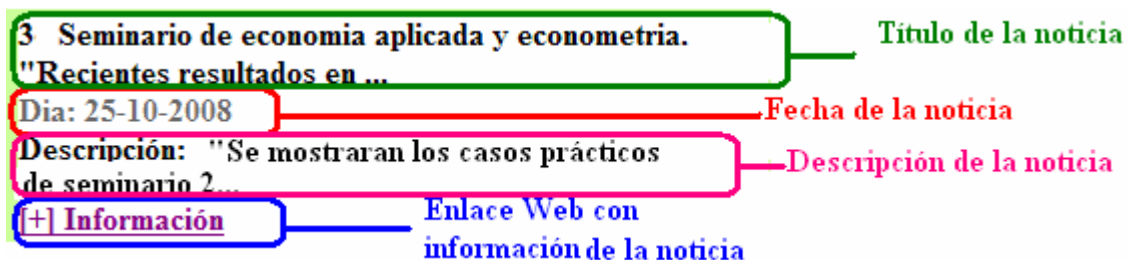


Figura A1- 6: Noticia

A1.3.3 INFORMACIÓN GENERAL DEL DEPARTAMENTO

Este apartado corresponde con la opción de menú de *¿Quiénes somos?*. En este enlace encontraremos información general del departamento, junto con una foto de grupo del mismo y un enlace a videos de interés.

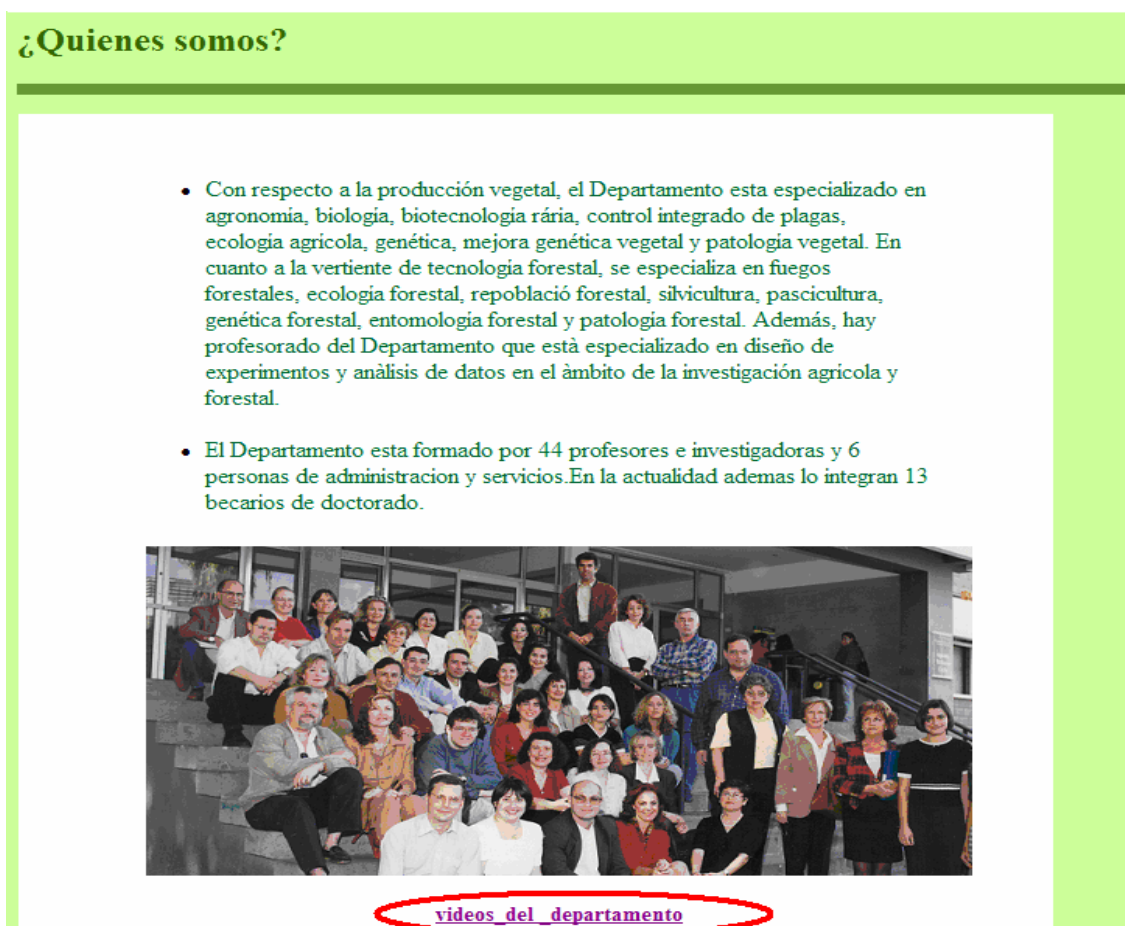


Figura A1- 7: Contenido de quiénes somos

A1.3.4 MIEMBROS DEL DEPARTAMENTO

La información acerca de los miembros que lo forman se ha clasificado en tres grupos: PAS, Becarios y PDI. Para ello tenemos en la opción de *Personal* un menú desplegable para que se pueda elegir la opción correspondiente:

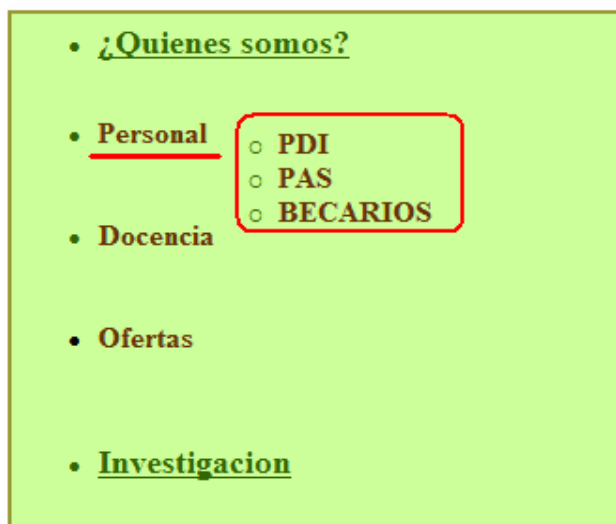


Figura A1- 8: Menú principal

Si seleccionamos la opción de *Personal*, nos mostrará información del personal del departamento:

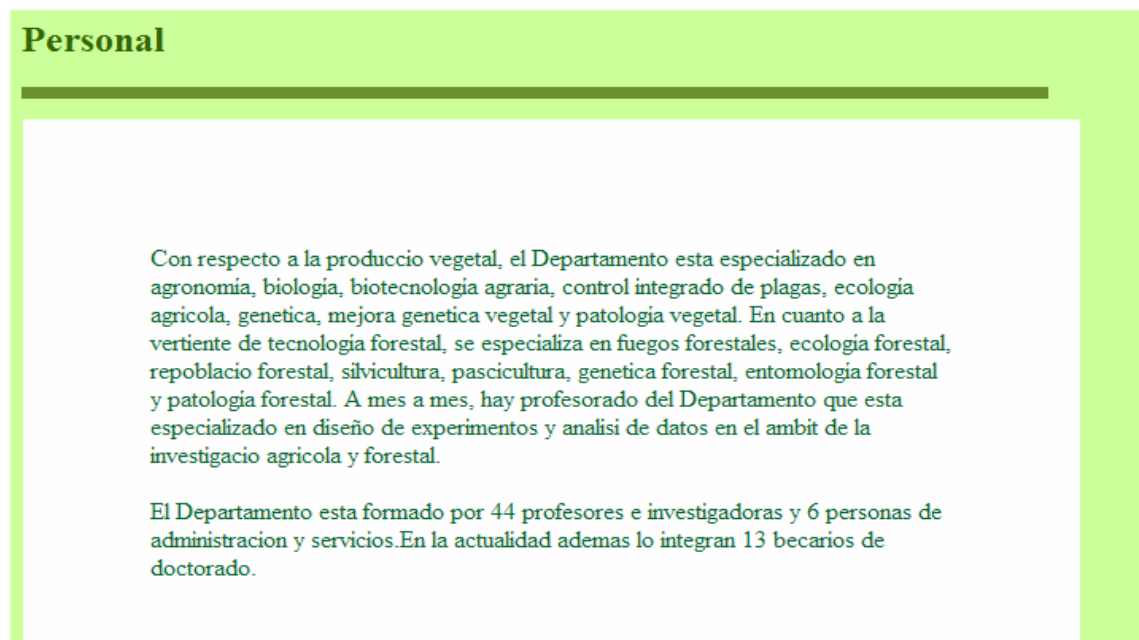


Figura A1- 9: Contenido personal



Para el resto de opciones, en cada grupo aparece un listado con los miembros que lo forman, estos tienen un enlace para poder acceder a sus currículos. Por ejemplo si seleccionamos **PDI**:

Página Personal PDI

Catedráticos

Fernandez Lopez, Pedro

Robles Cordero, Marta

Garcia Contreras, Jose

Coronado Gui, Ana

Carba Pelae, Roberto

Curículo

Titulares

Peman Baena, Estefania

Roche Fernandez, Monica

De la Paz Gutiérrez, Paloma

Sanchez Martinez Alvaro

Sanchez Rodriguez, Daniel

Ayudantes

Hugo Sanchez, Victor

Ruiz Rodriguez, Manuel

Moreno Diez, Lorena

Castillo Perez, Alejandra

Duelo Jimeno, Sara

Figura A1- 10: Personal PDI

A1.3.5 DOCENCIA DEL DEPARTAMENTO

Con las opciones que aparecen en el menú *desplegable de docencia* se podrán seleccionar, las *licenciaturas*, *masteres* y *cursos de especialización* que imparte el departamento. Para ello se desplegarán las opciones pasando el ratón por la zona de **Docencia**.

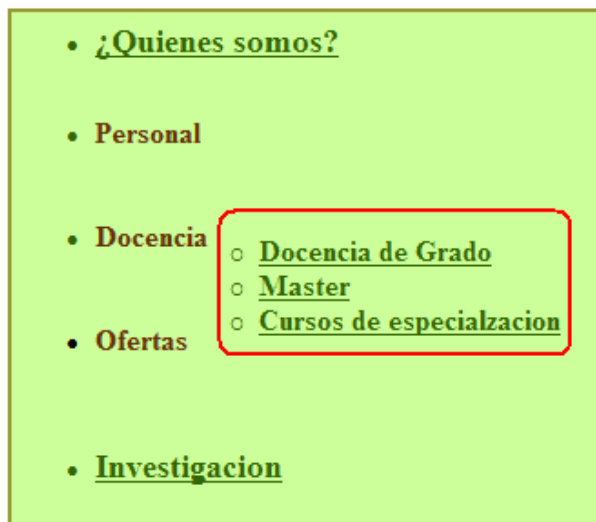


Figura A1- 11: Menú principal. Docencia

En cada una de las opciones se podrá acceder a:

- Si es una licenciatura, se puede acceder a los dosieres electrónicos de la asignatura. Para ello se debe pulsar en *Ver dossier-electrónico*, que contiene la información en la Web o a *Información general* que contendrá esta información en formato *pdf*.
- Si es un master, se puede acceder a más información del master pulsando sobre el mismo.
- Si es un curso de especialización, se puede acceder a más información del curso pulsando sobre el mismo.

A continuación se muestra un ejemplo de una licenciatura que aparece con todas las asignaturas que deben cursarse:

Docencia

Docencia de Grado

Enginyeria Tècnica Agrícola - Especialitat Explotacions Agropecuàries








- Càlcul - [Informacion general](#) 
Ver Dossier-electronic
- Mecànica - [Informacion general](#) 
Ver Dossier-electronic
- Química General - [Informacion general](#) 
Ver Dossier-electronic
- Biología - [Informacion general](#) 
Ver Dossier-electronic
- Topografía i Cartografía - [Informacion general](#) 
Ver Dossier-electronic
- Estadística i àlgebra - [Informacion general](#) 
Ver Dossier-electronic
- Electromagnetisme i Termodinàmica - [Informacion general](#) 
Ver Dossier-electronic

Figura A1- 12: Ejemplo licenciatura

A1.3.6 OFERTAS DEL DEPARTAMENTO

Con las opciones que aparecen en el menú *desplegable de ofertas* se podrán seleccionar, las becas, proyectos finales de carrera (PFC) y trabajos dirigidos que ofrece el departamento. Para ello se desplegarán las opciones pasando el ratón por la zona de *ofertas*.

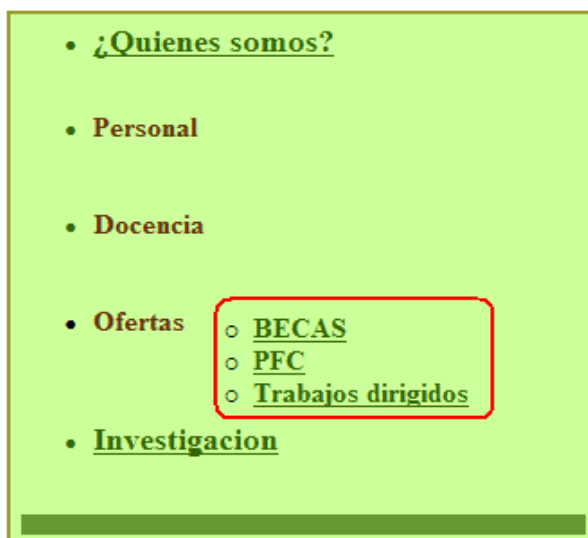


Figura A1- 13: Menú principal. Ofertas

A1.3.7 INVESTIGACIÓN

Seleccionando la opción de *Investigación* del menú principal se va a la página en la que se detallan la trayectoria profesional del grupo de investigación de ese departamento. Ofrece la posibilidad de poder acceder a esta información profesional: Grupos de investigación, Tesis doctorales y Transferencia Tecnológica.

Investigación

- El Departamento de PVCF ha tenido un fuerte compromiso con la investigación desde sus inicios, como lo demuestra su participación en más de 75 proyectos de investigación nacionales o en más de 20 proyectos europeos. Además de colaborar con numerosos centros de investigación nacionales o extranjeros, profesores e investigadores del Departamento integran diferentes centros de investigación:

• Fundació Centre UdL-IRTA (Institut de Recerca i Tecnologia Agroalimentàries)
Centre Tecnològic Forestal de Catalunya
Parc Científic i Tecnològic Agroalimentari

El resultado de la investigación son más de 60 patentes, casi 900 publicaciones en revistas, de ellas más de 200-300 en revistas SCI, más de 400 contribuciones en libros, más de 160 contratos con empresas o instituciones públicas y más de tesis leídas. Esta actividad supone una generación de recursos en actividades de I+D+I de más de 1 M€ al año.

Fruto de este compromiso con la investigación, a los profesores Ignacio Romagosa (1998) y Ramón Albajes (2002) se les concedió la Medalla Narcís Monturiol al mérito científico y tecnológico otorgada por la Generalitat de Catalunya.

Grupos de investigación
Tesis doctorales
Transferencia de tecnología

Figura A1- 14: Contenido investigación

En cada una de las opciones se podrá acceder a:

- Si es un grupo de investigación, se puede acceder a los grupos en sí, pulsando sobre el nombre de cada grupo.
- Si es una tesis, se puede acceder a información más detallada de la misma, seleccionando sobre ella.

Por ejemplo, se muestran los *grupos de investigación*, lo que está recuadrado en rojo indica que pulsando sobre él se puede acceder a una página con más información.

Investigación

Grupos de investigación

Grupos consolidados Generalitat

- Agronomía y calidad de cultivos extensivos
Linea de investigacion: linea_investigacion1
- Biotecnología vegetal aplicada
Linea de investigacion: linea_investigacion1

Grupos consolidados UdL

- Mejora genética de cultivos
Linea de investigacion: linea_investigacion1

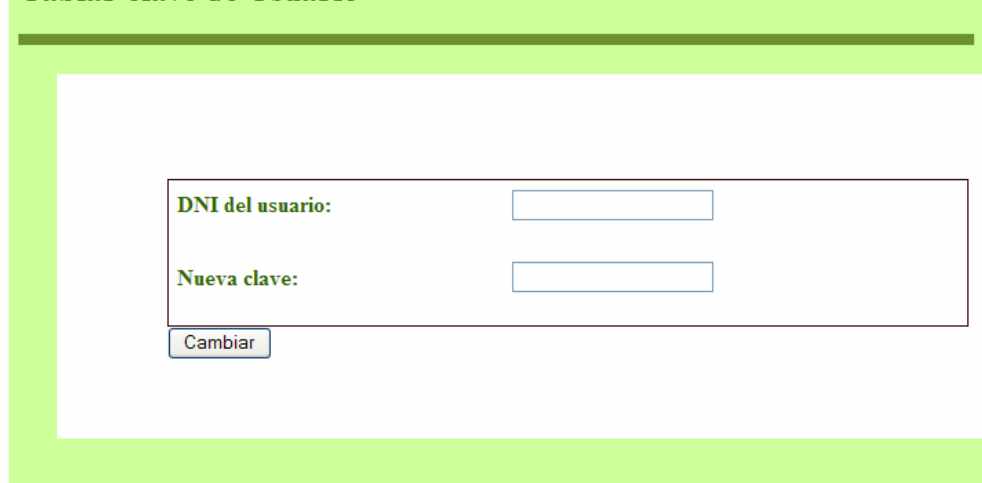
Figura A1- 15: Ejemplos Grupos de Investigación

En la en menú de la página principal del portal Web existe la opción de *Cambiar contraseña de usuario*, pero para poder realizarla el usuario debe de estar dado de alta en el portal Web.

A1.3.8 CAMBIAR LA CONTRASEÑA DE USUARIO

Si es un usuario registrado por la aplicación, es decir que pertenece al grupo del Departamento y ha sido dado de alta en esta aplicación. En el caso de no estar dado de alta en la aplicación, el usuario no podrá cambiar ninguna clave y recibirá un mensaje de error.

Cabiar clave de Usuario

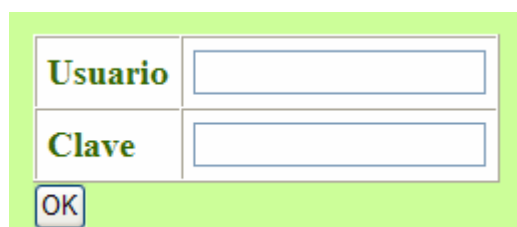


The form is titled 'Cabiar clave de Usuario' (Note the typo 'Cabiar'). It contains two input fields: 'DNI del usuario:' and 'Nueva clave:'. Below these fields is a button labeled 'Cambiar'.

Figura A1- 16: Cambio clave usuario

A1.3.9 VALIDAR USUARIO

Si se es un usuario registrado y se quiere entrar en la aplicación para realizar otras funciones, será necesario validarse.



The form is titled 'Validación usuario'. It contains two input fields: 'Usuario' and 'Clave'. Below these fields is a button labeled 'OK'.

Figura A1- 17: Validación usuario

En la casilla de usuario, se deberá introducir el DNI del usuario y en la de contraseña la registrada por el propio usuario.

En caso de realizar las operaciones correctamente, se pasa a otra página donde se da la bienvenida al usuario que abre la sesión y en caso de ser erróneo, aparecerá un mensaje de error.

A1.4 NAVEGAR COMO USUARIO REGISTRADO

Vamos a detallar las funcionalidades que puede realizar un usuario registrado en la aplicación.

A1.4.1 VALIDAR USUARIO

Previamente se ha de validar el usuario y dar un resultado positivo mostrando la página principal de las operaciones que pueden realizar los usuarios registrados. En ella, se **saluda** al usuario y siempre se mostrará la opción de “**desconectar**”:



Gestión

- Documentos
- Reservas
- Usuario
- Agenda
- Noticia

Agenda

Día: 2008-10-25 hora: 13:00 Segundo desayuno coloquio sobre conciliacion laboral y familiar [+] Información	Día: 2008-10-25 hora: 13:00 Segundo desayuno coloquio sobre conciliacion laboral y familiar [+] Información
Día: 2008-10-25 hora: 14:00 Conferencia sobre los nuevos planes de estudios: Bolonia [+] Información	Día: 2008-10-25 hora: 14:00 Seminario de economia de la empresa. Francisco Javier Husillos [+] Información

Noticias

1_ Seminario de economia aplicada y econometria. "Recientes resultados en Día: 20-10-2008 Descripción: El catedrático Roberto Berino [+] Información	2 Seminario de economia aplicada y econometria. "Recientes resultados en ... Día: 25-10-2008 Descripción: "Se conoceran los fundamentos básicos.... [+] Información
3_ Seminario de economia aplicada y econometria. "Recientes resultados en ... Día: 25-10-2008 Descripción: "Se mostraran los casos prácticos de seminario 2... [+] Información	4 Seminario de economia aplicada y econometria. "Recientes resultados en Día: 25-10-2008 Descripción: Se conoceran los fundamentos básicos... [+] Información

[Volver a la pagina principal](#)

Bienvenido/a: Marta Lopes Romero  [Desconectar](#)

Figura A1- 18: Pantalla principal usuario registrado

Todas las operaciones que se pueden realizar son:

- Documentos: añadir, ver documentos de actas y de consejos.
- Reservas: añadir, borrar y modificar reservas de coches, ordenadores, laboratorios y cañones.

- Usuario: modificar los datos (todos los usuarios pueden), borrar y añadir (sólo el administrador).
- Agenda: añadir información, modificar y borrar eventos privados o públicos.
- Noticia: añadir información, modificar y borrar noticias privadas o públicas.
- Volver a la página principal: El usuario registrado podrá volver a la página principal de usuarios registrados en todo momento.
- Desconectar: si el usuario desea salir de sus sesión, sólo tendrá que “**Desconectar**” y volverá a la página de principal de la aplicación.

Al desplazarse con el ratón en las distintas operaciones de menú, aparecerán desplegables detallando las operaciones.

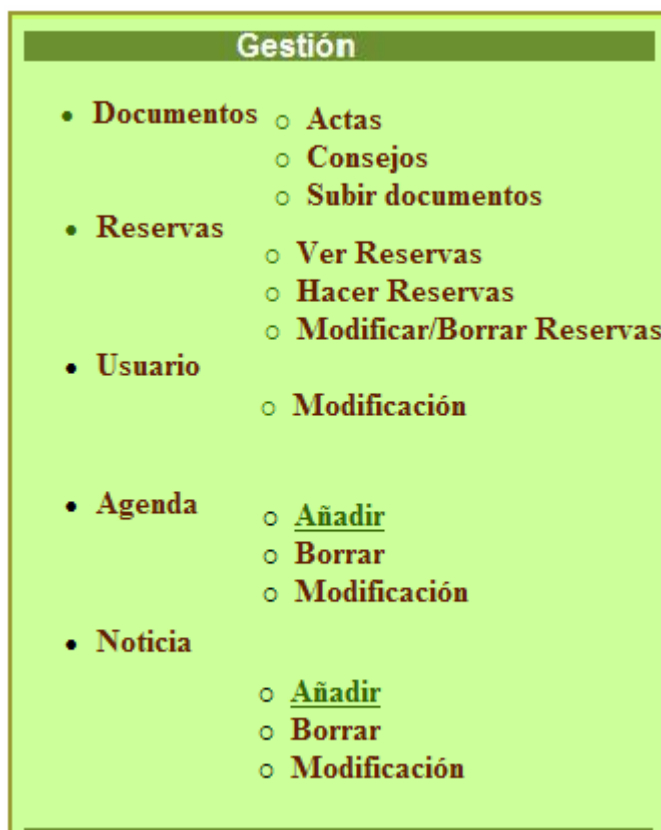


Figura A1- 19: Operaciones del menú de las páginas de Usuario Registrados

A1.4.2 VISUALIZAR DATOS DE LA AGENDA Y DE NOTICIAS INTERNAS.

La información de las noticias y eventos del departamento que son privados se muestran en la página principal de la aplicación cuando el usuario se ha validado.



The screenshot shows a user dashboard for 'Marta Lopes Romero'. At the top right, there is a 'Desconectar' link. The main content area is divided into two sections: 'Agenda' and 'Noticias'. The 'Agenda' section lists two events for October 25, 2008: 'Segundo desayuno coloquio sobre conciliacion laboral y familiar' and 'Seminario de economia de la empresa. Francisco Javier Husillos'. Each event has a '[+] Información' link. The 'Noticias' section lists four news items, all titled 'Seminario de economia aplicada y econometria. "Recientes resultados en ..."', with dates of October 20 or 25, 2008, and descriptions of 'contenido'. Each news item also has a '[+] Información' link. The 'Agenda' and 'Noticias' section headers are highlighted with red boxes in the original image.

Bienvenido/a: Marta Lopes Romero [Desconectar](#)

Agenda

Dia: 25-10-2008
Segundo desayuno coloquio sobre conciliacion laboral y familiar
[\[+\] Información](#)

Dia: 25-10-2008
Seminario de economia de la empresa. Francisco Javier Husillos
[\[+\] Información](#)

Noticias

1. Seminario de economia aplicada y econometria. "Recientes resultados en ..."
Dia: 20-10-2008
Descripción: contenido
[\[+\] Información](#)

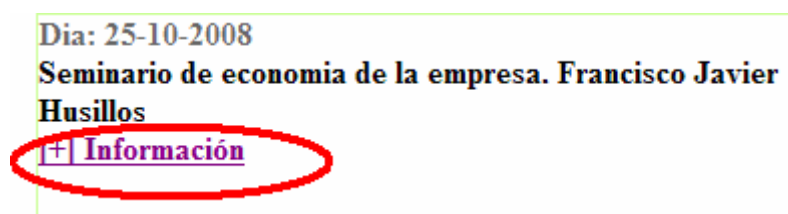
2. Seminario de economia aplicada y econometria. "Recientes resultados en ..."
Dia: 25-10-2008
Descripción: contenido
[\[+\] Información](#)

3. Seminario de economia aplicada y econometria. "Recientes resultados en ..."
Dia: 25-10-2008
Descripción: contenido
[\[+\] Información](#)

4. Seminario de economia aplicada y econometria. "Recientes resultados en ..."
Dia: 25-10-2008
Descripción: contenido
[\[+\] Información](#)

Figura A1- 20: Pantalla principal usuarios registrados. Agenda

Para obtener una información mas detallada, se deberá seleccionar en “[+] información” y se abrirá una página Web que contiene información más detallada:



The screenshot shows a detailed view of a seminar. It displays the date 'Dia: 25-10-2008', the title 'Seminario de economia de la empresa. Francisco Javier Husillos', and a description 'Descripción: contenido'. A red oval highlights the '[+] Información' link at the bottom of the content area.

Dia: 25-10-2008
Seminario de economia de la empresa. Francisco Javier Husillos
[\[+\] Información](#)

Figura A1- 21: Más información

A1.4.3 VISUALIZAR DOCUMENTOS: ACTAS O CONSEJOS.

Esta información se obtiene pasando el ratón por encima de la opción **documentos** y pinchando en lo que se desea ver.



Figura A1- 22: Menú de usuarios registrados. Gestión de documentos

A1.4.4 SUBIR DOCUMENTOS: ACTAS O CONSEJOS

Esta información se obtiene pasando el ratón por encima del título documentos y pinchando en la opción *subir*.

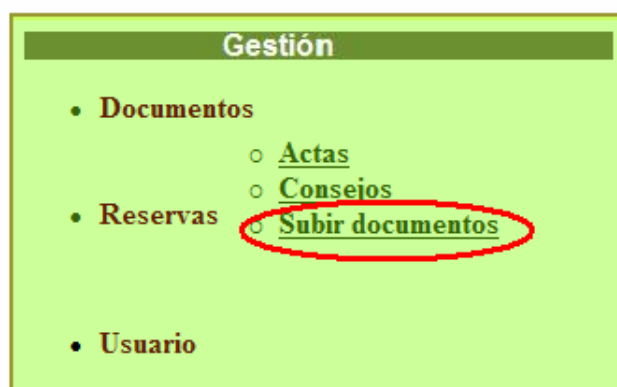



Figura A1- 23: Menú usuarios registrados. Subir documentos

Se deberá rellenar el formulario y si la validación de los campos rellenados es correcta, se grabarán los datos. El usuario debe **rellenar todos los campos**.

Subir Documento

[] Nombre:	<input type="text"/>
Tipo del documento :	ACTAS <input type="button" value="v"/>
RUTA :	<input type="text"/> <input data-bbox="1029 627 1157 660" type="button" value="Examinar..."/>
URL :	<input type="text"/>
<input type="button" value="OK"/>	

Ayuda

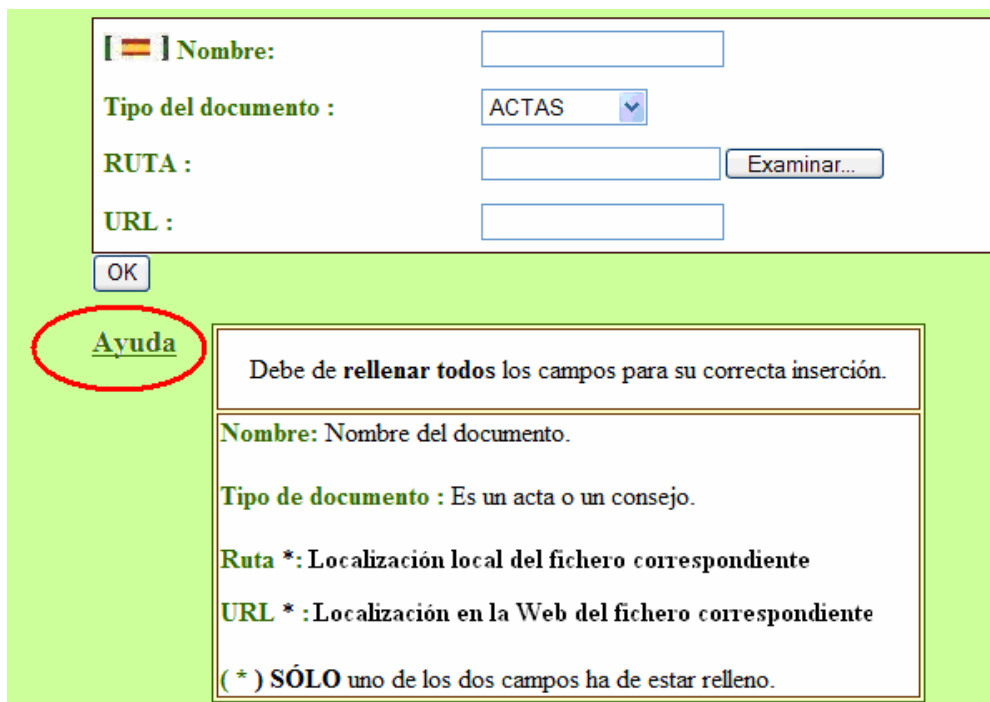
Figura A1- 24: Subir documentos. Campos

Los campos que debe rellenar el usuario en el formulario son:

- **Nombre del documento:** título con el que se desea mostrar en la Web.
- **Tipo de documento:** si es un *acta* o un *consejo*.
- A elegir **uno de estos dos campos:** Ruta o URL:
 - **RUTA:** Es el lugar físico del documento, el lugar donde se encuentra en el ordenador.
 - **URL:** dirección Web en la que está subido el documento. Debe empezar por “<http://>”.

En el caso de que **no se rellenen todos los campos**, en concreto Nombre del documento y URL, **y se haya seleccionado el botón adjuntar**, la aplicación estará esperando a que los rellene y no hará nada más.

Además se incluye una sección de **Ayuda** para solucionar las posibles dudas del usuario. Se accede pasando el “ratón” sobre la palabra *ayuda* que se encuentra en el formulario.



The image shows a web form for uploading documents. The form has the following fields: a language selector (Spanish flag), a text input for 'Nombre:', a dropdown menu for 'Tipo del documento:' with 'ACTAS' selected, a text input for 'RUTA:', a text input for 'URL:', and an 'Examinar...' button. Below the form is an 'OK' button. To the left of the form is a red circle containing the word 'Ayuda'. To the right of the 'OK' button is a help dialog box with the following text: 'Debe de rellenar todos los campos para su correcta inserción.', 'Nombre: Nombre del documento.', 'Tipo de documento : Es un acta o un consejo.', 'Ruta *: Localización local del fichero correspondiente', 'URL * : Localización en la Web del fichero correspondiente', and '(*) SÓLO uno de los dos campos ha de estar relleno.'

Figura A1- 25: Formulario subir documentos

A1.4.5 VER RESERVAS

El usuario puede ver las reservas que se han hecho a partir de la fecha actual. Para ello selecciona la opción *ver reservas*.

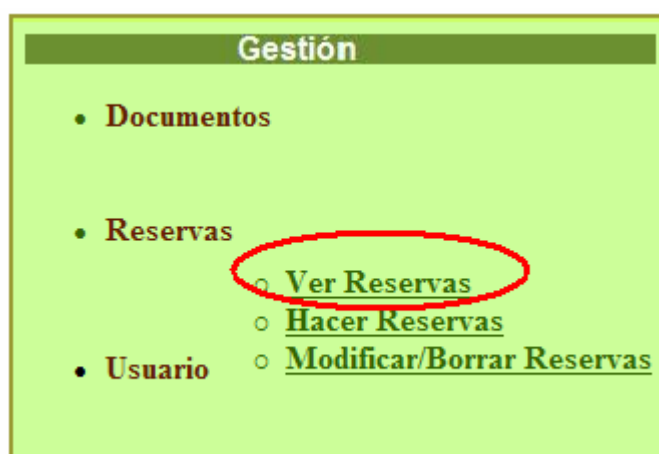


Figura A1- 26: Menú usuarios registrados. Visualización reservas

Esta opción le lleva a una página con la posibilidad de elegir entre ver reservas de coches, ordenadores, cañones o laboratorios. El usuario elegirá una de estas cuatro opciones a visualizar.



Figura A1- 27: Menú ver reservas

Las reservas que se mostrarán serán a partir de la fecha actual en la que nos encontremos.
Por ejemplo si seleccionamos en ver reservas ordenador, aparecerá una lista con todas ellas:

Reservas

Ordenador

Identificador de la reserva: 1	
Fecha: 2008-10-28	Hora: 14:00 - 15:00
Usuario: Marta Lopes Romero	
Identificador Ordenador: 1	
Identificador de la reserva: 2	
Fecha: 2008-10-27	Hora: 14:00 - 15:00
Usuario: Sara Ortega Baena	
Identificador Ordenador: 3	
Identificador de la reserva: 3	
Fecha: 2008-10-25	Hora: 14:00 - 18:00
Usuario: Marta Lopes Romero	
Identificador Ordenador: 1	
Identificador de la reserva: 4	
Fecha: 2008-10-25	Hora: 14:00 - 18:00

Figura A1- 28: Reservas de ordenadores

A1.4.6 HACER RESERVAS

El usuario puede realizar reservas, para ello debe seleccionar la opción *Hacer Reservas* que aparece al pasar el “ratón” sobre la palabra “reservas” del menú principal. Aparecerá un formulario con las cuatro posibles reservas, pero **sólo** se puede rellenar y realizar **una** reserva de las cuatro posibilidades mostradas. Para poder realizar una segunda, el usuario deberá de realizar la misma operación que para la primera, seleccionando del menú izquierdo la opción *hacer reserva...*



Figura A1- 29: Menú usuarios registrados. Hacer reservas

Cuando se llega a la página del formulario de reserva se deben rellenar todos los campos de la reserva que se pretende realizar. Este proceso se ha facilitado con campos que tienen un desplegable indicando las posibles opciones.

En la página que contiene los formularios para realizar las reservas aparece, en la parte superior, un enlace de **las 3 posibles reservas** a realizar con el fin de desplazarte al formulario indicado.

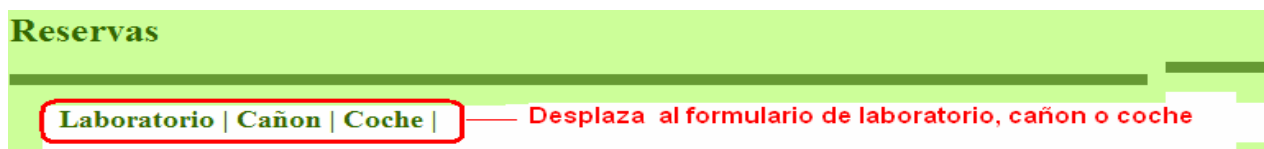


Figura A1- 30: Facilidad de acceso a formulario

Además se incluye una sección de **ayuda** en la parte inferior de la página con el objetivo de solucionar las posibles incertidumbres del usuario respecto a la reserva. En ella se define cada uno de los campos del formulario:

- **Día:** día para el que se quiere reservar.
- **Mes:** mes en el que se quiere realizar la reserva.
- **Año:** año en el que se quiere hacer la reserva.
- **Duración:** duración de la reserva.
- **Modelo:** modelo de lo que se quiere reservar.
- **Número:** identificador del objeto a reservar.

Reservas

Laboratorio | Cañon | Coche |

Ordenadores					
Día	01	Mes	Enero	Año	
Hora	9:00	Duración	1 hora	Modelo	portatil
Número	1	<input type="button" value="Reservar"/>			

Laboratorio					
Día	01	Mes	Enero	Año	
Hora	9:00	Duración	1 hora		
Número	1	<input type="button" value="Reservar"/>			

Cañon					
Día	01	Mes	Enero	Año	
Hora	9:00	Duración	1 hora		
Número	1	<input type="button" value="Reservar"/>			

Coche					
Día	01	Mes	Enero	Año	
Hora	9:00	Duración	1 hora	Modelo	rojo
Número	1	<input type="button" value="Reservar"/>			

[Ayuda](#)

Figura A1- 31: Formulario Reservas

Si el usuario ha realizado con éxito la reserva, le mostrará una pantalla con los datos de las reservas realizadas en la aplicación. En caso contrario, pueden ocurrir dos cosas:

1. Si **no ha rellenado todos los datos** del formulario (en concreto el campo del **año**), no hará nada, mostrará el mismo formulario a rellenar.
2. Si se ha intentado reservar algo que **ya estaba reservado** se mostrará un mensaje de error, indicando que no se puede hacer la reserva.

A1.4.7 MODIFICAR RESERVAS

El usuario puede modificar las reservas que ha realizado. Pero **sólo** podrá realizar la **modificación de la fecha** de la reserva de las reservas que haya creado previamente. Para ello, deberá seleccionar esta opción del menú:



Figura A1- 32: Menú modificar reservas

A continuación, mostramos la página que contiene las opciones de modificar o borrar reservas. (Se muestra una vez seleccionado la opción de “**Modificar/Borrar Reservas**”)

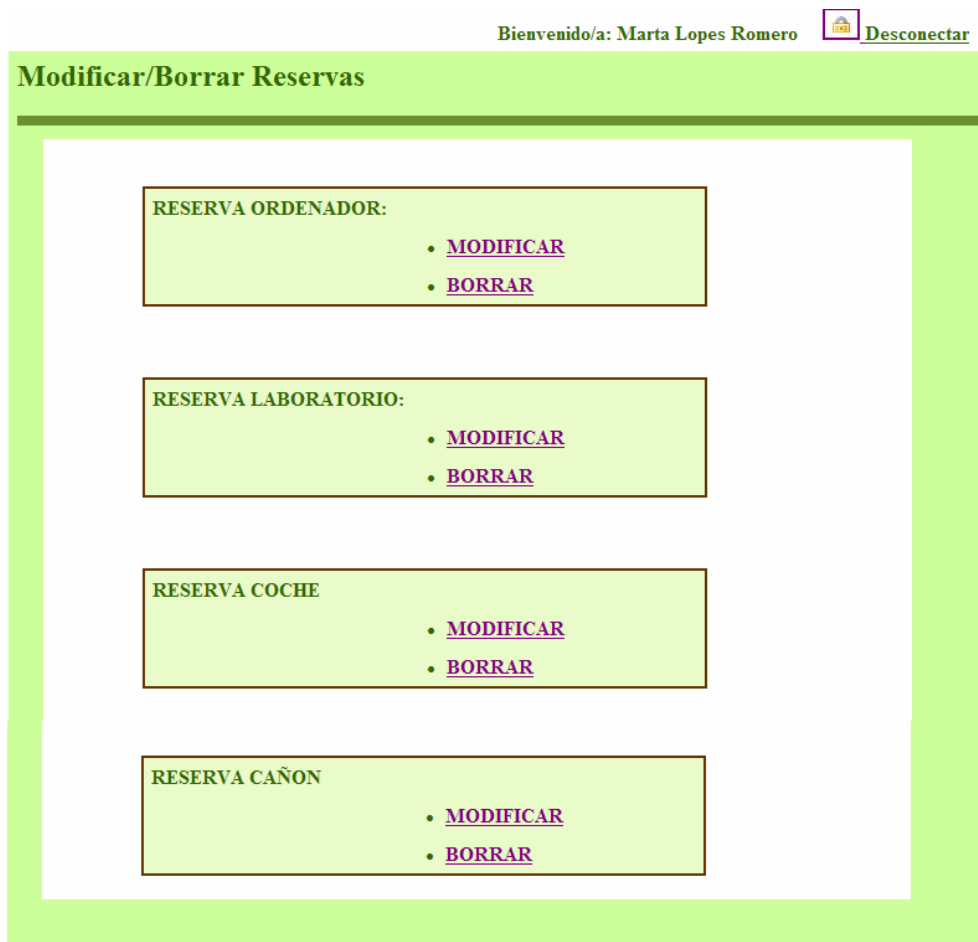


Figura A1- 33: Modificar o borrar reservas

Una vez seleccionada la opción de “**modificar o borrar**”, mostrará todas las reservas que existen posteriores a la fecha actual, por ejemplo, seleccionamos **modificar reservas de ordenadores** y aparece la lista de reservas de ordenadores con un “**check**” para que se seleccione la reserva que se desea modificar. Se muestra a continuación:

Modificar Ordenador

Identificador de la reserva: 1	<input type="radio"/>
Fecha: 2008-10-28	Hora: 14:00 - 15:00
Usuario: Marta Lopes Romero	
N ° de Ordenador: 1	
Identificador de la reserva: 2	<input type="radio"/>
Fecha: 2008-10-27	Hora: 14:00 - 15:00
Usuario: Sara Ortega Baena	
N ° de Ordenador: 1	
Identificador de la reserva: 3	<input type="radio"/>
Fecha: 2008-10-25	Hora: 14:00 - 18:00
Usuario: Marta Lopes Romero	
N ° de Ordenador: 1	
Identificador de la reserva: 4	<input checked="" type="radio"/>
Fecha: 2008-10-25	Hora: 14:00 - 18:00
Usuario: Sara Ortega Baena	
N ° de Ordenador: 2	

MODIFICAR

Figura A1- 34: Modificar reserva de ordenador

Seleccionamos una de ellas y si el **usuario** es el que ha **realizado la reserva** que se desea **modificar**, podrá modificarla. En el caso de que el **usuario no sea el mismo**, mostrará un **mensaje de aviso** en la parte central de la aplicación indicando que no la puede realizar, ya que no es suya. Ejemplo:

La operación NO SE PUEDE REALIZAR porque el usuario NO HA REALIZADO la reserva seleccionada , pertenece al usuario Sara Ortega Baena

Figura A1- 35: Mensaje de error (1)

En el caso de ser un usuario administrador, el administrador podrá modificar cualquier reserva, independientemente de que la haya realizado.

Una vez modificada la reserva, puede ocurrir:

1. Que se haya realizado con éxito, entonces, mostrará una lista con todas las reservas para que se puede verificar la modificación de la misma. Como la figura ver reserva de ordenadores, “Figura A1-28: Reserva de Ordenadores”.
2. Error, en el caso de que el usuario intente reservar en una fecha en la que ya exista una reserva. Recordamos que sólo se puede modificar las fechas de la reserva, ningún campo más.

A1.4.8 BORRAR RESERVAS

El usuario puede borrar las reservas que ha realizado. Para ello, deberá seleccionar esta opción del menú:

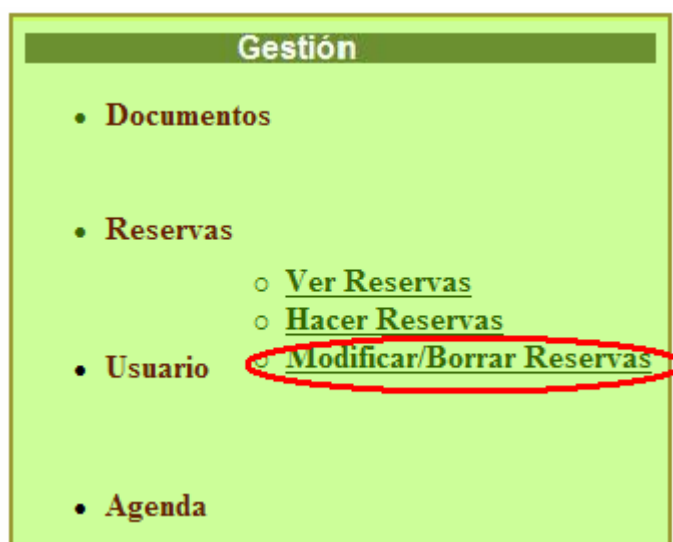


Figura A1- 36: Menú borrar o modificar reservas

A continuación, mostramos la página que contiene con las opciones de menú tras seleccionar la opción de “**Modificar/Borrar Reservas**”.

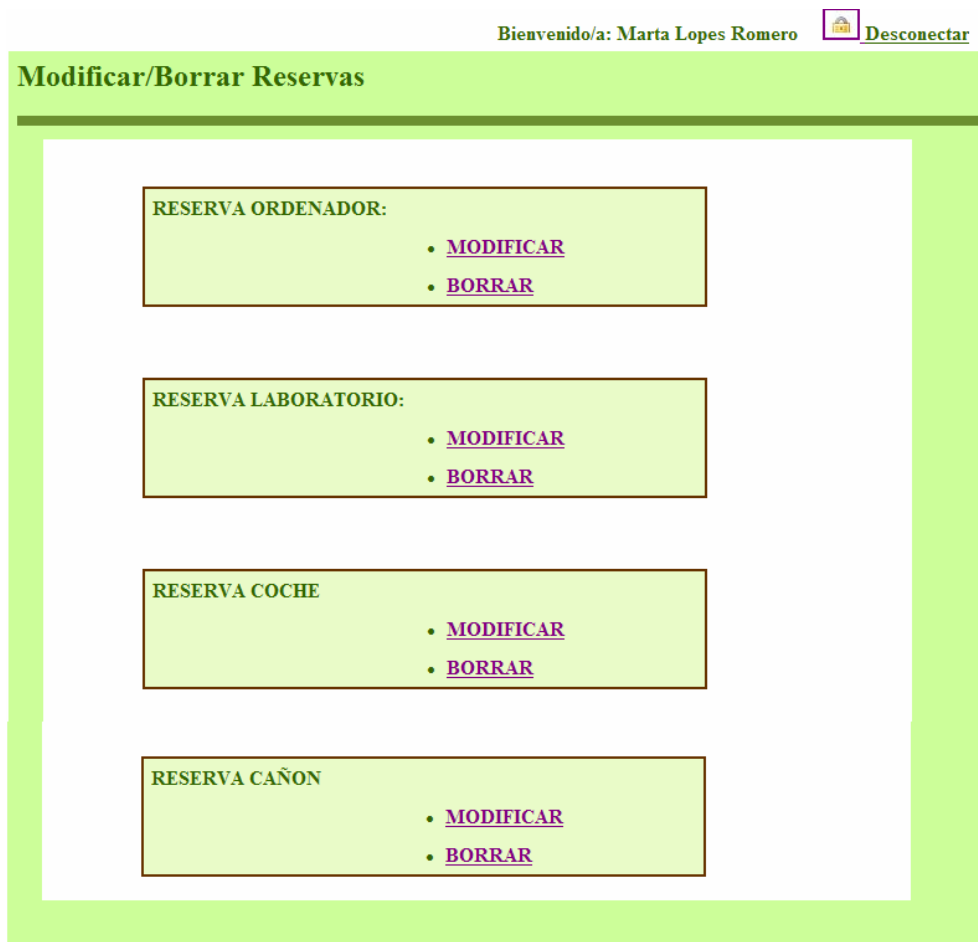


Figura A1- 37: Borrar o Modificar Reservas

Una vez seleccionada la opción de “**Borrar**” una reserva, mostrará todas las reservas que existen posteriores a la fecha actual, por ejemplo, seleccionamos **Borrar reservas de ordenadores** y aparece la lista de reservas de ordenadores con un “**check**” para que se seleccione una de ellas a modificar. Se muestra a continuación:

Borrar Reserva de Laboratorio

Identificador de la reserva: 1	<input type="radio"/>
Fecha: 2008-10-25	Hora: 14:00 - 18:00
Usuario: Marta Lopes Romero	
N ° de Laboratorio: 1	
Identificador de la reserva: 2	<input checked="" type="radio"/>
Fecha: 2008-10-25	Hora: 14:00 - 18:00
Usuario: Sara Ortega Baena	
N ° de Laboratorio: 2	

BORRAR

Figura A1- 38: Borrar Reserva Laboratorio

Seleccionamos una de ellas y si el **usuario** es el que ha **realizado la reserva** que se desea **borrar**, podrá borrarla. En el caso de que el **usuario no sea el mismo**, mostrará un **mensaje de aviso** en la parte central de la página, indicando que no la puede realizar, ya que no es suya. Ejemplo:

La operación **NO SE PUEDE REALIZAR** porque el usuario **NO HA REALIZADO** la reserva seleccionada , pertenece al usuario Sara Ortega Baena

Figura A1- 39: Mensaje de error (2)

En el caso de ser un usuario administrador, el administrador podrá borrar cualquier reserva, independientemente de que la haya realizado.

Una vez seleccionada la reserva y pulsado el botón “Borrar”, puede ocurrir:

1. Que se haya realizado con éxito, entonces, mostrará un mensaje de aviso en la parte central de la página indicando el éxito. En dato “Nº” del mensaje corresponde con el identificador numérico de la reserva.

SE HA BORRADO LA RESERVA Nº 1 con fecha de reserva 2008-10-25 realizada por el usuario Marta Lopes Romero

Figura A1- 40: Mensaje de éxito


2. Error, en el caso de que el usuario intente borrar una reserva que no sea suya, se le mostrará el mensaje de aviso en la parte central de la página:

La operación NO SE PUEDE REALIZAR porque el usuario NO HA REALIZADO la reserva seleccionada , pertenece al usuario Sara Ortega Baena


Figura A1- 41: Mensaje de error (3)

A1.4.9 ALTA USUARIO

Esta tarea es del **usuario administrador**. Si el administrador quiere dar de alta un usuario en la aplicación deberá acceder a este apartado del menú. Y aparecerá la página con el siguiente formulario:

Bienvenido/a: Nieves Peman Baena  [Desconectar](#)

Alta Usuario

Nombre del usuario:	<input type="text"/>
Apellidos:	<input type="text"/>
Tipo del usuario :	<div>PAS  PAS PDI-Catedratico PDI-Ayudante PDI-Titular Administrador BECARIO</div>
DNI:	<input type="text"/>
Añadir clave:	<input type="text"/>
URL:	<input type="text"/>

[Ayuda](#)

Figura A1- 42: Campos alta usuario

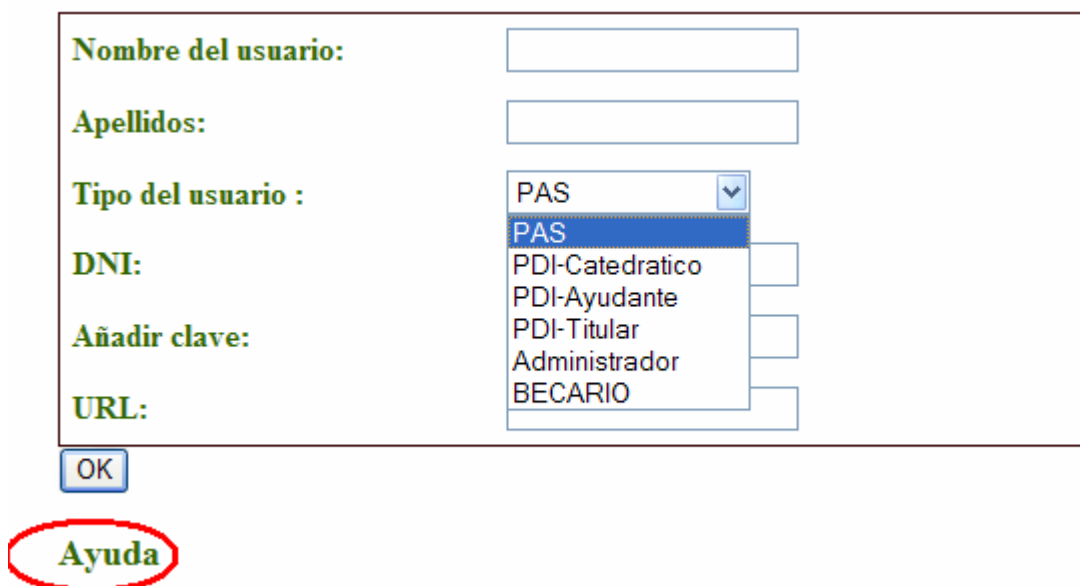
Para poder dar un alta en la aplicación, se deberán rellenar los datos del nuevo usuario del departamento, para ello se muestra un formulario con los siguientes campos:

- **Nombre usuario:** Nombre del nuevo miembro.
- **Apellidos:** Apellidos del nuevo miembro.
- **Tipo de usuario:** si pertenece a PDI-Catedrático, PDI-Ayudante, PDI-Titular, PAS, BECARIO o ADMINISTRADOR.

- **DNI:** el DNI del miembro.
- **Añadir clave:** se pone una clave para que pueda acceder a la aplicación.
- **URL:** Dirección Web donde se encuentra su currículum.

Se deben rellenar todos los campos o la aplicación mostrará un mensaje de error. En el caso de haberse rellenado con éxito, se volverá a la página principal de los usuarios registrados.

Para poder satisfacer las posibles dudas del usuario en cuanto al alta se refiere se proporciona una opción de **ayuda** situada en la parte inferior del formulario. La ayuda contiene la definición de los campos del formulario que hemos comentado anteriormente:



The image shows a web form for user registration. It contains the following fields and controls:

- Nombre del usuario:** A text input field.
- Apellidos:** A text input field.
- Tipo del usuario :** A dropdown menu with the following options: PAS (selected), PAS, PDI-Catedratico, PDI-Ayudante, PDI-Titular, Administrador, and BECARIO.
- DNI:** A text input field.
- Añadir clave:** A text input field.
- URL:** A text input field.
- OK:** A button to submit the form.
- Ayuda:** A button located below the form, circled in red, which provides help for the registration process.

Figura A1- 43: Opción de ayuda en altas usuarios

A1.4.10 MODIFICAR USUARIO

Cada usuario podrá modificar sus datos personales y a parte, el usuario administrador podrá modificar cualquier dato de los usuarios.

Para modificar los datos, sólo se requiere el identificador de usuario con el que se accede a la aplicación, éste es el DNI del mismo.

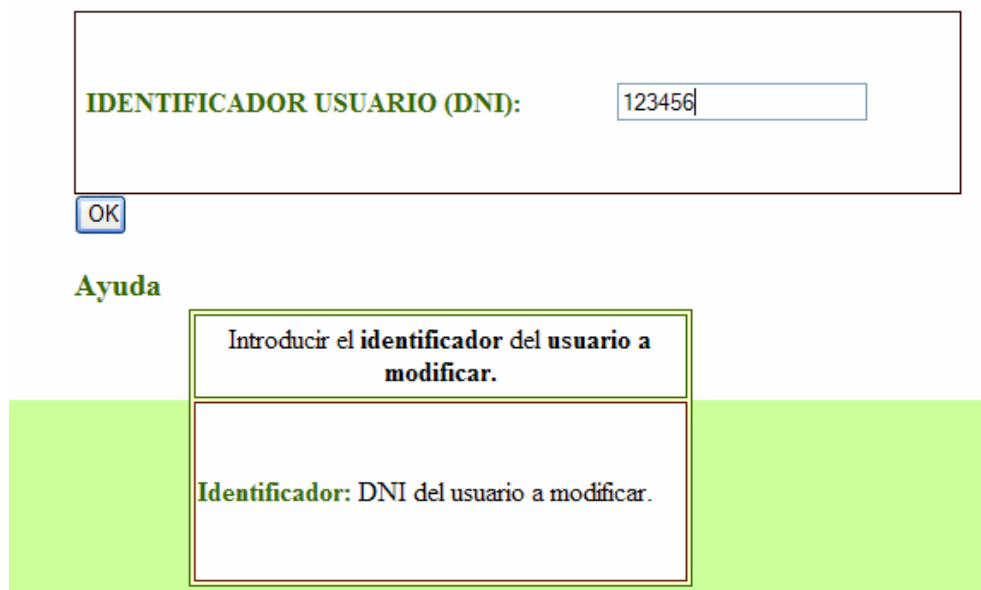


Figura A1- 44: Introducir DNI-1º paso de Modificación Usuario

Si se introducen mal los datos del DNI, el sistema mostrará un mensaje indicando que el usuario con ese DNI no existe.



Figura A1- 45: Mensaje de error (4). Del 1º paso Modificación Usuario

En caso contrario mostrará la información del usuario para poderla modificar:

Nombre del usuario:	<input type="text" value="Marta"/>
Apellidos:	<input type="text" value="Lopes Romero"/>
Tipo del usuario :	<input type="text" value="PAS"/>
DNI:	<input type="text" value="123456"/>
Añadir clave:	<input type="text" value="nini"/>
URL:	<input type="text" value="marta.lopez@udl.es"/>

Ayuda

Figura A1- 46: Formulario relleno-2º paso de Modificación de Usuario

En el caso de ser **usuario administrador**, tendrá que introducir el DNI del usuario que desee modificar. No obstante, para este usuario, se muestra una opción de **Ayuda** que le facilita la tarea de modificar usuario.

A1.4.11 BAJA USUARIO

Esta acción es propia del **usuario administrador** podrá borrar usuarios del sistema.

Primero deberá de seleccionar la opción del menú de *baja*:



Figura A1- 47: Operación Baja de usuario

A continuación aparecerá una pantalla para que introduzca el DNI del usuario que se desea dar de baja:

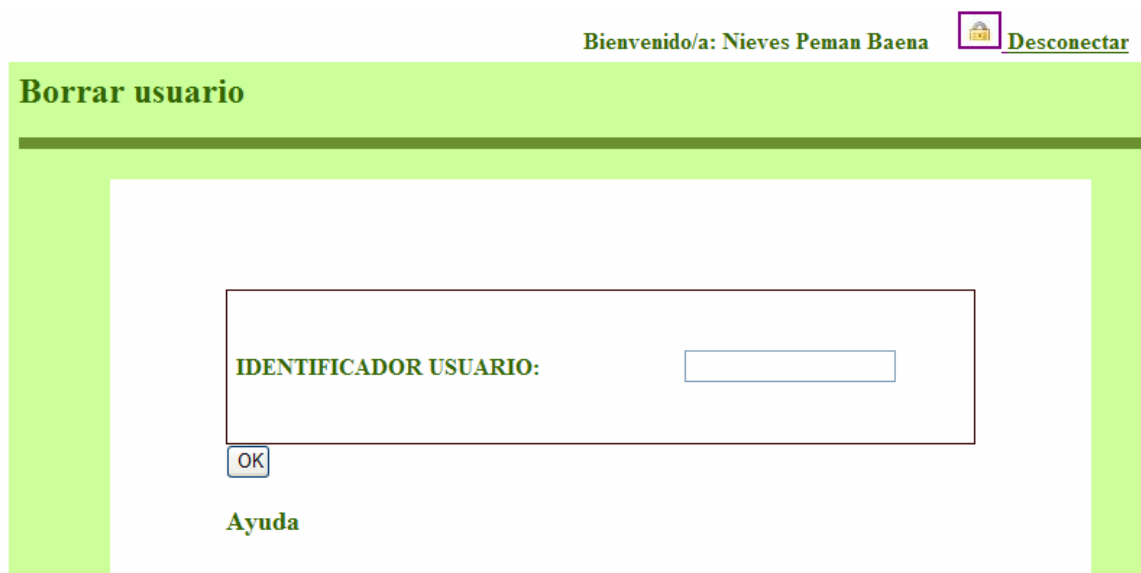


Figura A1- 48: Borrar usuario

Como se observa en la figura “Borrar usuario”, en la parte inferior tiene una opción de **Ayuda** en la que se indica que información ha de introducir el usuario asociado a “Identificador Usuario”.

A1.4.12 AÑADIR NOTICIAS

La información que se muestra en esta sección se asocia a noticias que han ocurrido o que han publicado los profesores, por ejemplo, si ya han salido los listados de prácticas del curso o unas ofertas de becas o si se ha publicado las actas de notas.

La información que se muestra en esta sección será la que corresponde al mes en curso y se ha desarrollado para que muestre sólo 4 noticias en esta sección, ya que así se indicaba en los requisitos.

Para poder añadir noticias, el usuario tendrá que seleccionar el apartado del menú principal con el nombre de “**Añadir noticias**”. Para realizar correctamente la inserción, el usuario tendrá que rellenar la información en los 3 idiomas, que se le indica mediante mensajes de aviso. La

noticia se añade primero en el idioma en el que se está navegando la aplicación y después se añade en los restantes idiomas.

Para poder insertar noticias, el usuario deberá rellenar el formulario:

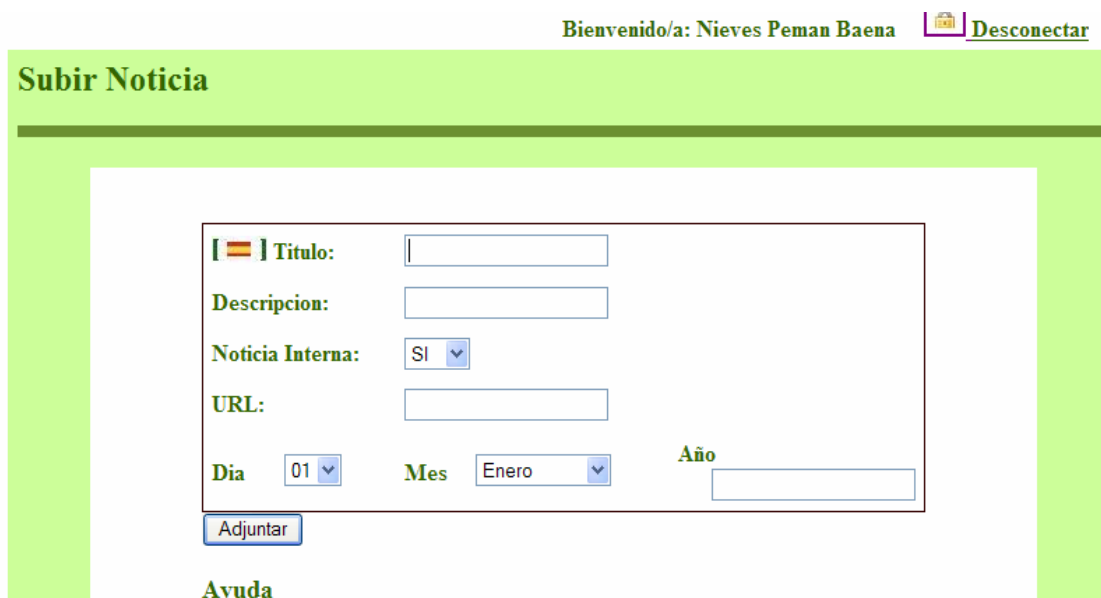


Figura A1- 49: Subir noticia

Como se puede ver en la “Figura A1-46: Subir noticia”, aparece la opción de “Ayuda” en la que se define cada campo del formulario:

- **Título:** Título de la información.
- **Información interna:** Se refiere a si la información que se quiere mostrar, se quiere mostrar en la zona pública de Agenda o en la de privada. Si se selecciona SI, se mostrará en la zona de privada para que sólo la puedan ver los miembros del departamento. En caso contrario, se mostrará la información sólo en la zona pública
- **URL:** Hace referencia a la dirección en la cual esta disponible la información. Y contiene todos los datos de la misma.
- **Descripción:** Descripción de la noticia
- **Día:** Día en el que se produce la noticia.
- **Mes:** Mes en el que se producirá el noticia o en la que se ha subido.
- **Año:** Año en el que se produce la noticia. Tiene que ser numérico de 4 dígitos, ej.: 2008

En el caso de **NO RELLENAR** todos los campos *Título, Descripción, URL y, año* si se selecciona la opción *Adjuntar*, **no se añadirá la noticia**.

Como hemos comentado anteriormente, para realizar correctamente la inserción, el usuario tendrá que rellenar la información en los 3 idiomas, se indica al mismo mediante mensajes de aviso. La noticia se añade primero en el idioma en el que se está navegando la aplicación y después se añade en los restantes idiomas. Por ejemplo, estamos en el idioma castellano y hemos rellenado el formulario mostrado previamente, nos mostraría:

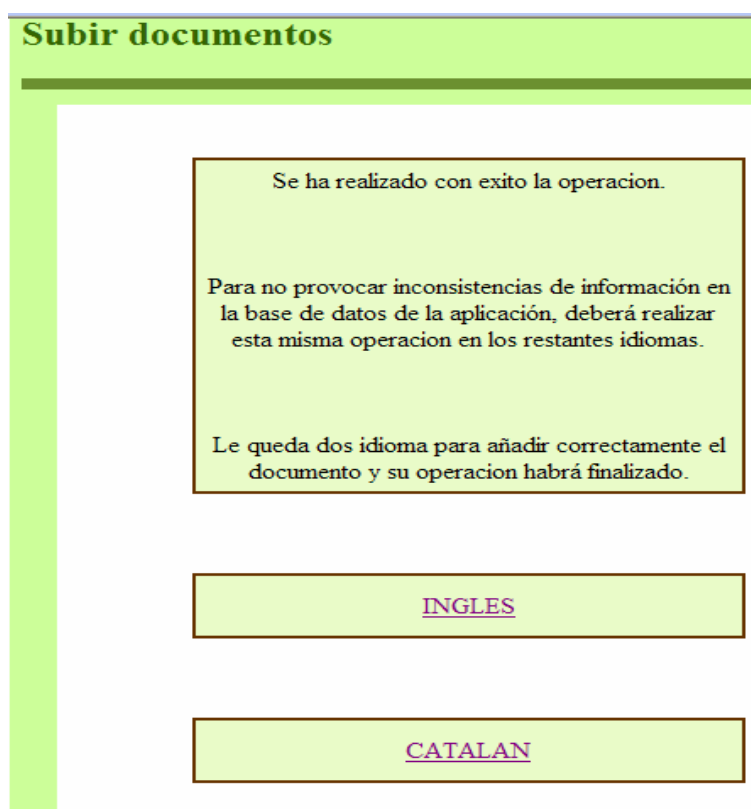


Figura A1- 50: Rellenar noticia Idiomas

Cada vez que se rellene el formulario en un idioma, se mostrara una pantalla similar a la figura anterior indicando el idioma que le falta por añadir.

Si se ha realizado con éxito la inserción en todos los idiomas, se mostrará la página principal de la aplicación de usuarios registrados.

A1.4.13 MODIFICAR NOTICIAS

Para poder modificar noticias, el usuario tendrá que seleccionar en menú principal de los usuarios registrados la opción “**Modificación**”.

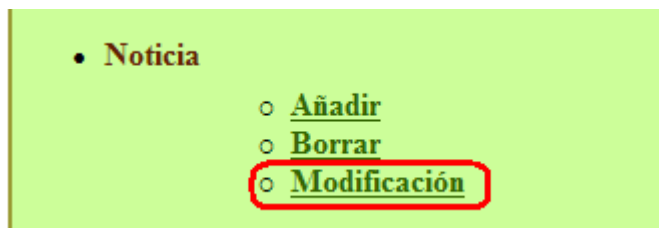


Figura A1- 51: Menú modificar noticia

A continuación, se mostrarán una página intermedia en la que el usuario debe seleccionar si desea modificar las noticias que son públicas al departamento o las que son privadas.

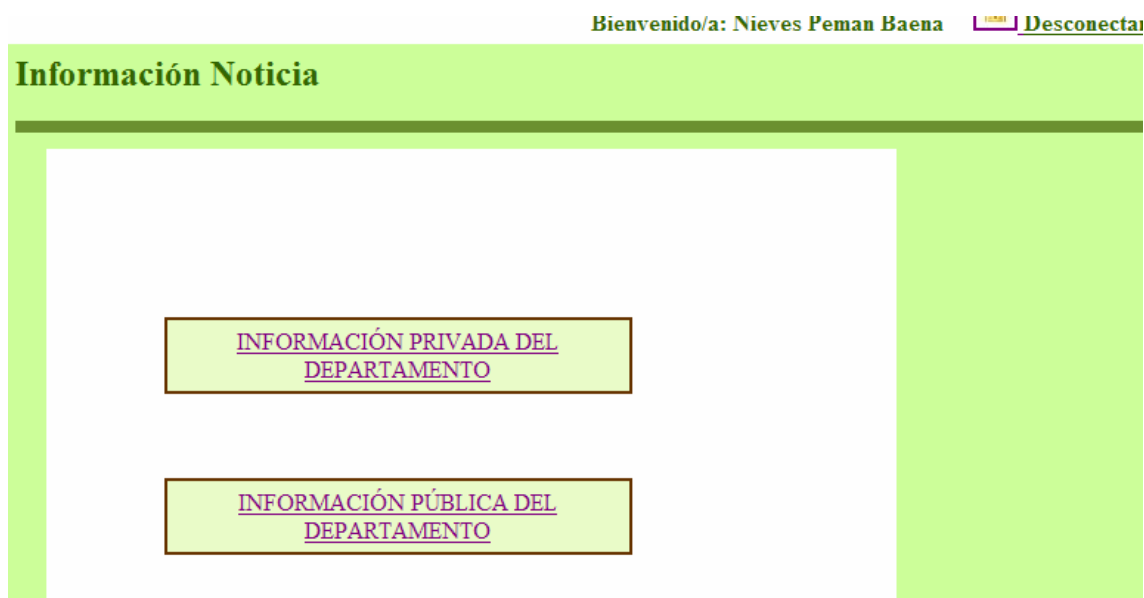


Figura A1- 52: Información pública o privada

Después el usuario seleccionará uno de éstos enlaces y se mostrará todas las noticias públicas o privadas para que se pueda seleccionar la que se quiere modificar.

Modificar Información Noticia

Identificador de la noticia: 5	<input type="radio"/>
Fecha: 2008-10-20	
Descripción: "Se conocerán los fundamentos básicos...."	
Título: 1 Seminario de economía aplicada y econometría. "Recientes resultados en"	
Interna: si	
URL: http://www.uc3m.es	
Identificador de la noticia: 6	<input checked="" type="radio"/>
Fecha: 2008-10-25	
Descripción: " Debate entre los mejores alumnos de..."	
Título: 2 Seminario de economía aplicada y econometría. "Recientes resultados en"	
Interna: si	
URL: http://www.uc3m.es	

MODIFICAR

Figura A1- 53: Modificar Noticia

Una vez seleccionada la noticia que se desea modificar, se muestra un formulario con los datos de la noticia para facilitar su modificación.

Realizada la modificación, mostrará la página principal de usuarios registrados de la aplicación, “Figura A1-18: Pantalla principal usuario registrado”.

A1.4.14 BORRAR NOTICIAS

Para poder borrar noticias, el usuario tendrá que seleccionar en el menú principal de la página de usuarios registrados la opción “Borrar”.

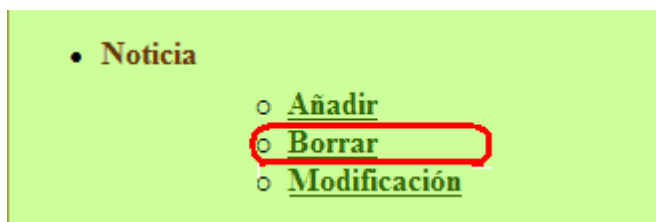


Figura A1- 54: Menú borrar noticia

A continuación, se mostrarán una página en la que el usuario debe seleccionar si desea borrar las noticias que son públicas al departamento o las que son privadas.

Información Noticia

[INFORMACIÓN PRIVADA DEL
DEPARTAMENTO](#)

[INFORMACIÓN PÚBLICA DEL
DEPARTAMENTO](#)

Figura A1- 55: Información noticia privada o pública

Después el usuario seleccionará uno de éstos enlaces y se mostrará todas las noticias públicas o privadas para que se pueda seleccionar la que se quiere borrar. Y una vez seleccionada la noticia, se borra y se muestra la pantalla principal de usuarios restringidos, “Figura A1-18: Página principal usuario registrado”.

Borrar Noticia

Identificador de la noticia: 5	<input type="radio"/>
Fecha: 2008-10-20	
Descripción: "Se conoceran los fundamentos básicos...."	
Título: 1 Seminario de economía aplicada y econometría. "Recientes resultados en"	
Interna: si	
URL: http://www.uc3m.es	
Identificador de la noticia: 6	<input checked="" type="radio"/>
Fecha: 2008-10-25	
Descripción: " Debate entre los mejores alumnos de..."	
Título: 2 Seminario de economía aplicada y econometría. "Recientes resultados en ..."	
Interna: si	
URL: http://www.uc3m.es	

BORRAR

Figura A1- 56: Seleccionar noticia para borrar

A1.4.15 AÑADIR INFORMACIÓN AGENDA

La información que se añade en la sección de agenda corresponde a los posibles eventos que pueden tener los profesores y/o los alumnos, por ejemplo una reunión general del departamento o una fiesta de inauguración de curso por parte de la escuela. Esta información se muestra en la página principal del portal y en la página principal de los usuarios que se han registrado.

La aplicación mostrará la información de esta sección si el evento tiene como fecha, desde la actual hasta un mes posterior a la misma. Se mostrarán como máximo cuatro 4 eventos en esta sección ya que así se definió en los requisitos.

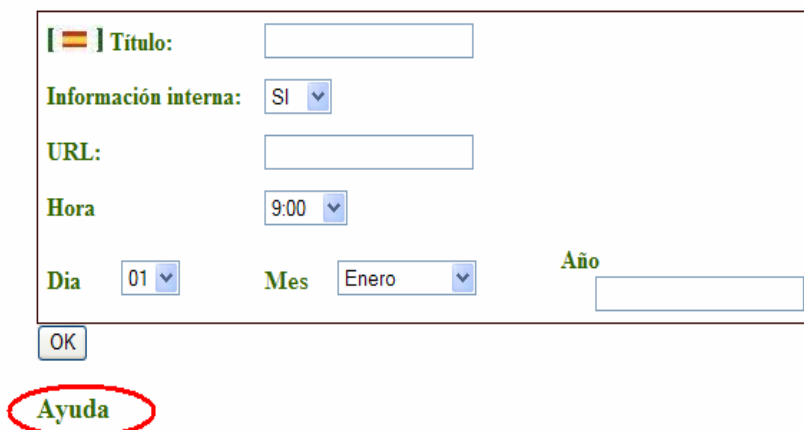
Para realizar esta operación, seleccionamos:




Figura A1- 57: Añadir Evento en la Agenda

A continuación, se mostrará un formulario para rellenar los datos de la misma. Se incorpora una opción de **Ayuda** en la que mostrará la definición de los campos a comentar.

Subir información Agenda



[] Título:

Información interna: SI

URL:

Hora: 9:00

Día: 01 Mes: Enero Año:

[Ayuda](#)

Figura A1- 58: Añadir información agenda

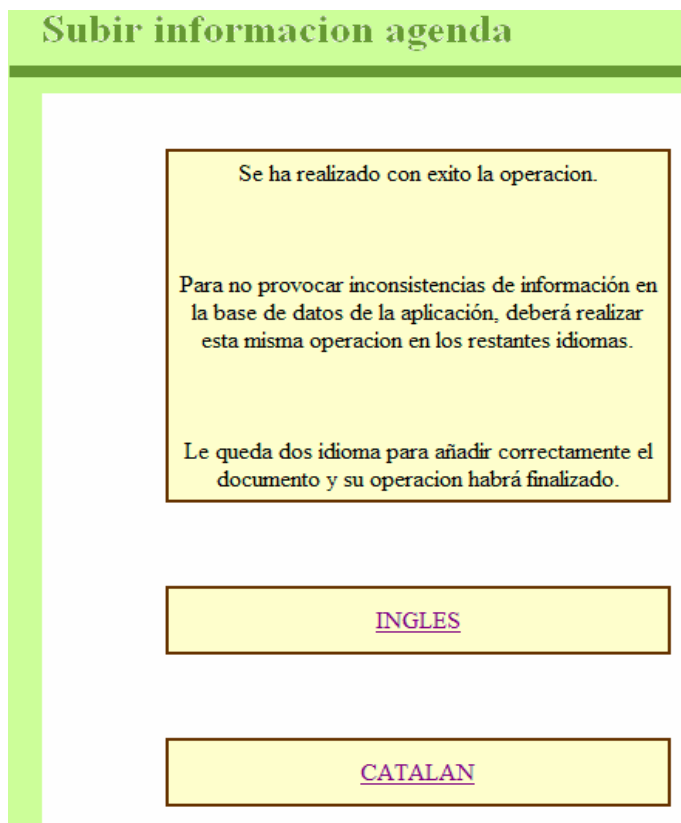
Cada campo del formulario significa:

- **Título:** Título de la información.
- **Información interna:** Se refiere a si la información que se quiere mostrar, se quiere mostrar en la zona pública de Agenda o en la de privada. Si se selecciona SI, se mostrará en la zona de privada para que sólo la puedan ver los miembros del departamento. En caso contrario, se mostrará la información sólo en la zona pública
- **URL:** Hace referencia a la dirección en la cual está disponible la información.
- **Hora:** Hora en la que se producirá el evento.
- **Día:** Día en el que se produce el evento.
- **Mes:** Mes en el que se produce el evento.
- **Año:** Año en el que se produce el evento. Tiene que ser en formato numérico de 4 dígitos.

En el caso de **NO RELLENAR** los campos *Título*, *URL* y, *año* si se selecciona la opción *Adjuntar*, **no añadirá** el evento a la aplicación.

Para completar la operación de poder añadir información agenda, el usuario tendrá que rellenar la información en los **3 idiomas (catalán, español e inglés)**, la cual se indica al mismo

mediante mensajes de aviso. Es decir, la información se añade primero en el idioma en el que se está navegando por la aplicación y después se añade con los restantes idiomas. Por ejemplo, después de rellenar el formulario anterior que está en **español**, aparecerá la siguiente pantalla, en la que muestra que debe rellenar la información en 2 idiomas más:



Subir informacion agenda

Se ha realizado con exito la operacion.

Para no provocar inconsistencias de información en la base de datos de la aplicación, deberá realizar esta misma operacion en los restantes idiomas.


Le queda dos idioma para añadir correctamente el documento y su operacion habrá finalizado.

[INGLES](#)


[CATALAN](#)

Figura A1- 59: Selección idiomas restantes en añadir evento

El usuario deberá de seleccionar el **inglés** o el **catalán** y se mostrará un formulario con la información en el idioma seleccionado, por ejemplo si selecciona el **inglés**, el portal Web mostrará la siguiente pantalla:

Bienvenido/a: Marta Lopes Romero  [Desconectar](#)

Añadir informacion Agenda

 Title:

Information:

YES

URL:

Hour

9:00

Day

27


Month

Octubre

Year

Figura A1- 60: añadir información de la agenda (Inglés)

Todos los campos, excepto el título, aparecerán insertados para facilitar al usuario esta inserción en el **inglés**.

Cuando rellene el campo título y seleccione el botón  , se mostrará la pantalla que indica el/los idiomas restantes para insertar la agenda, en este caso sólo queda el **catalán**:

Subir informacion agenda

Se ha realizado con exito la operacion.

Para no provocar inconsistencias de información en la base de datos de la aplicación, deberá realizar esta misma operacion en los restantes idiomas.

Le queda un idioma para añadir correctamente el documento y su operacion habrá finalizado.

[CATALAN](#)

Figura A1- 61: Información Agenda Idioma Catalán

Cómo muestra la figura anterior, al usuario le queda por rellenar la información en catalán, deberá seleccionar el catalán y después el portal Web mostrará el formulario asociado para que lo rellene. Es similar al formulario de la “Figura A1-60” pero relleno.

Una vez relleno, se ha realizado con éxito la operación y se mostrará la página principal de la aplicación de usuarios registrados, en dónde se podrá visualizar la información si la fecha del evento es posterior o igual a la actual.

A1.4.16 MODIFICAR INFORMACIÓN AGENDA

Para poder modificar la información de la agenda, el usuario tendrá que seleccionar el apartado “Modificar” del menú principal de las páginas de usuarios registrados.

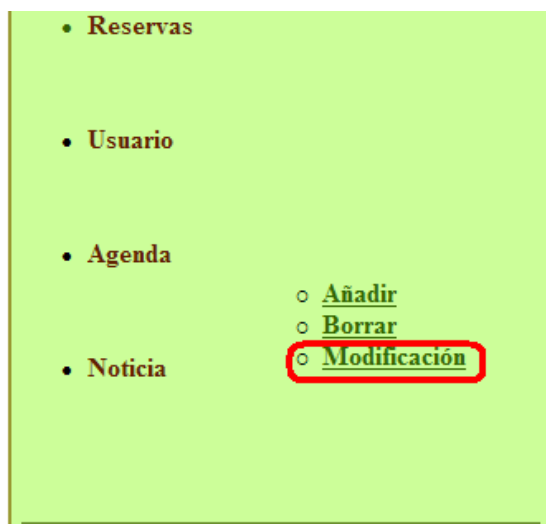


Figura A1- 62: Modificar agenda

En primer lugar, el usuario deberá de elegir si desea modificar información privada o pública del departamento en la pantalla que se muestra a continuación:

Información Agenda

[INFORMACIÓN PRIVADA DEL
DEPARTAMENTO](#)

[INFORMACIÓN PÚBLICA DEL
DEPARTAMENTO](#)

Figura A1- 63: Información agenda pública o privada

Una vez seleccionada una opción, mostrará todas las noticias que existen de acuerdo con la selección. En este caso se seleccionó la información pública del Departamento:

Agenda

Modificar Información Agenda

Identificador de la noticia: 5	<input type="radio"/>
Fecha: 2008-10-25	Hora: 13:00
Titulo: Segundo desayuno coloquio sobre conciliación laboral y familiar	
URL: http://www.uc3m.es	
Identificador de la noticia: 6	<input type="radio"/>
Fecha: 2008-10-25	Hora: 14:00
Titulo: Seminario de economía de la empresa. Francisco Javier Husillos-Carqu?s.r	
URL: http://www.uc3m.es	
Identificador de la noticia: 10	<input type="radio"/>
Fecha: 2008-10-27	Hora: 09:00
Titulo: Nueva información agenda	
URL: http://www.udl.es	

Figura A1- 64: Lista de eventos públicos

De esta lista de eventos seleccionamos el que queramos modificar y se muestra un formulario con los datos de la noticia para facilitar su modificación.

Por ejemplo si queremos modificar la noticia nº 5:



Figura A1- 65: Modificar agenda

Una vez mostrado este formulario, se puede modificar cualquier dato, y cómo se necesita mantener la consistencia en la Base de datos en los 3 idiomas, si se realiza cualquier cambio en la noticia, se modificará en los 3 idiomas, excepto el campo “**título**” que se modifican sólo en el idioma que se está navegando en ese momento.

Como se puede apreciar en la “Figura A1-65: Modificar Agenda” tiene la opción de ayuda para facilitar al usuario la operación. En ella se explicará que si se realiza cualquier cambio en la noticia, se modificará en los 3 idiomas, excepto el campo “título” que se modifican sólo en el idioma que se está navegando en ese momento.

A1.4.17 BORRAR INFORMACIÓN AGENDA

Para poder borrar la información de la agenda, el usuario tendrá que seleccionar el apartado del menú que indica “Borrar”.

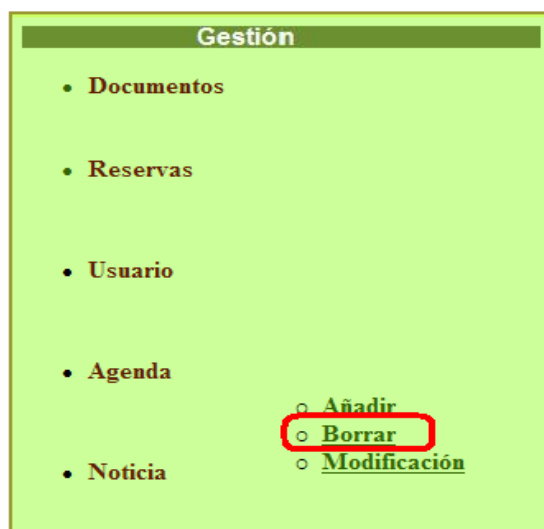


Figura A1- 66: Menú borrar agenda

A continuación el usuario deberá de seleccionar que tipo de información quiere borrar: información privada o pública:

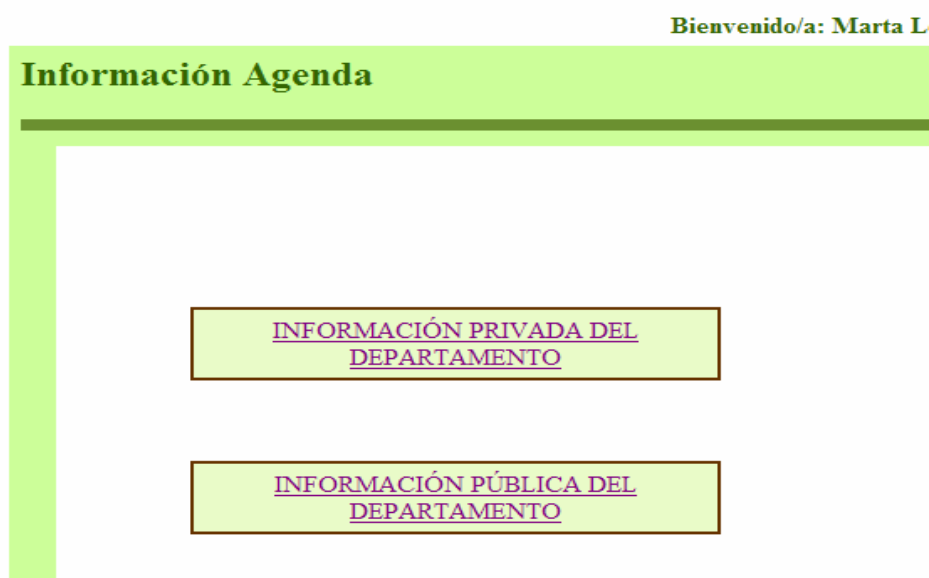


Figura A1- 67: Información agenda pública o privada

Después de seleccionar el tipo de información que se quiere seleccionar se mostrará una lista con la información de la agenda correspondiente con dicha selección.

Borrar Información Agenda

Identificador de la noticia: 5	<input type="radio"/>
Fecha: 2008-10-25	Hora: 13:00
Titulo: Segundo desayuno coloquio sobre conciliación laboral y familiar	
URL: http://www.uc3m.es	
Identificador de la noticia: 6	<input type="radio"/>
Fecha: 2008-10-25	Hora: 14:00
Titulo: Seminario de economía de la empresa. Francisco Javier Husillos	
URL: http://www.uc3m.es	
Identificador de la noticia: 10	<input type="radio"/>
Fecha: 2008-10-27	Hora: 09:00
Titulo: Nueva información agenda	
URL: http://www.udl.es	

BORRAR

Figura A1- 68: Borrar noticia

De esta lista de eventos se debe seleccionar el que se quiera borrar. Una vez seleccionado un evento y pulsado el botón “Borrar”, se borra y se muestra la página principal de usuarios registrados.

A1.4.18 VOLVER A LA PÁGINA PRINCIPAL DE USUARIOS REGISTRADOS

Si el usuario quiere volver a la zona registrada, lo hace pinchando en la opción de *Volver a página principal* que se encuentra en las opciones de menú de cualquier página de acceso a usuarios registrados.

• **Volver a la página principal**

Figura A1- 69: Volver a la página principal de usuarios registrados

A1.4.19 VOLVER A LA PÁGINA PRINCIPAL DE LA APLICACIÓN

Si el usuario quiere volver a la página principal de la aplicación, lo hace pinchando en la opción de “*Desconectar*”, que cierra la sesión del usuario y muestra la página principal de la aplicación. Esta opción se encuentra en la zona superior de todas las pantallas de acceso a usuarios registrados.



Figura A1- 70: Opción Desconectar



A1.5 CONFIGURACIÓN E INSTALACIÓN DEL PORTAL WEB

Para el correcto funcionamiento de la aplicación, tendremos que tener instalados los siguientes programas y comprobar que se cumplen los requisitos a continuación citados:

- El Sistema Operativo es independiente de la plataforma.
- Se debe tener instalado o instalar MySql versión 5.0 o superior para almacenar la base de datos que se va a usar en la aplicación.
- Necesitaremos tener instalado Java, con JDK 1.5.
- Se tendrá que instalar, si no se tiene ya instalado, el servidor Apache Tomcat para poder ejecutar la aplicación, ya que se trata de una aplicación Web.

Si ya se tiene instalado todo, hay que **tener en cuenta** que la **aplicación desarrollada**:

- Se debe ejecutar el script de creación de la Base de Datos. Se adjunta al final del manual.
- El fichero de configuración *hibernate.cfg.xml* tiene para la base de datos MySql el usuario *admin* y la contraseña *root*.

Si ya tiene instalada la Base de Datos MySql con credenciales de acceso diferentes a las especificadas en la aplicación:

- El fichero de configuración *hibernate.cfg.xml* en la zona de acceso a la base de datos MySql deberá contener el usuario y contraseña correspondientes a las credenciales de acceso a MySQL.



SCRIPT MySQL

```
CREATE TABLE TIPO_USUARIO(  
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    Nombre VARCHAR(50) NOT NULL  
  
    )TYPE=INNODB;  
CREATE TABLE USUARIO(  
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    Cargo INT NOT NULL,  
    Contraseña VARCHAR(100) NOT NULL,  
    Nombres VARCHAR(100) NOT NULL,  
    Apellidos VARCHAR(100) NOT NULL,  
    Tipo_miembro VARCHAR(100) NOT NULL,  
    DNI VARCHAR(100) NOT NULL,  
    URL VARCHAR(100) NOT NULL  
  
    )TYPE=INNODB;  
ALTER TABLE USUARIO ADD FOREIGN KEY (Cargo)  
    REFERENCES TIPO_USUARIO(id)  
    ON DELETE CASCADE;  
CREATE TABLE COCHE(  
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    Modelo VARCHAR(100) NOT NULL  
  
    )TYPE=INNODB;  
CREATE TABLE RESERVA_COCHE(  
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    id_coche INT NOT NULL,  
    id_usuario INT NOT NULL,  
    Fecha_reserva DATE NOT NULL,  
    Hora VARCHAR(50) NOT NULL,  
    Hora_f VARCHAR(50) NOT NULL  
  
    )TYPE=INNODB;  
ALTER TABLE RESERVA_COCHE ADD FOREIGN KEY (Id_usuario)  
    REFERENCES USUARIO(id)  
    ON DELETE CASCADE;  
ALTER TABLE RESERVA_COCHE ADD FOREIGN KEY (Id_coche)  
    REFERENCES COCHE(id)  
    ON DELETE CASCADE;  
CREATE TABLE LABORATORIO (  
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY  
  
    )TYPE=INNODB;  
CREATE TABLE RESERVA_LAB (  
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    id_lab INT NOT NULL,  
    id_usuario INT NOT NULL,  
    Fecha_reserva DATE NOT NULL,  
    Hora VARCHAR(50) NOT NULL,  
    Hora_f VARCHAR(50) NOT NULL  
  
    )TYPE=INNODB;
```



```
ALTER TABLE RESERVA_LAB ADD FOREIGN KEY (id_usuario)
REFERENCES USUARIO (id)
ON DELETE CASCADE;
ALTER TABLE RESERVA_LAB ADD FOREIGN KEY (id_lab)
REFERENCES LABORATORIO (id)
ON DELETE CASCADE;
CREATE TABLE ORDENADOR (
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    Modelo VARCHAR(100) NOT NULL
)TYPE=INNODB;
CREATE TABLE RESERVA_ORDENADOR (
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    id_ord INT NOT NULL,
    id_usuario INT NOT NULL,
    Fecha_reserva DATE NOT NULL,
    Hora VARCHAR(50) VARCHAR(50) NOT NULL,
    Hora_f VARCHAR(50) NOT NULL
)TYPE=INNODB;
ALTER TABLE RESERVA_ORDENADOR ADD FOREIGN KEY (Id_ord)
REFERENCES ORDENADOR (id)
ON DELETE CASCADE;
ALTER TABLE RESERVA_ORDENADOR ADD FOREIGN KEY (Id_usuario)
REFERENCES USUARIO (id)
ON DELETE CASCADE;
CREATE TABLE CANON (
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY
)TYPE=INNODB;
CREATE TABLE RESERVA_CANON (
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    id_can INT NOT NULL,
    id_usuario INT NOT NULL,
    Fecha_reserva DATE NOT NULL
)TYPE=INNODB;
ALTER TABLE RESERVA_CANON ADD FOREIGN KEY (Id_can)
REFERENCES CANON (id)
ON DELETE CASCADE;
ALTER TABLE RESERVA_CANON ADD FOREIGN KEY (Id_usuario)
REFERENCES USUARIO (id)
ON DELETE CASCADE;
CREATE TABLE ASIGNATURA (
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    Nombre VARCHAR(50) NOT NULL,
    Descripcion VARCHAR(50) NOT NULL,
    URL VARCHAR(50) NOT NULL
)TYPE=INNODB;
CREATE TABLE LICENCIATURA(
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    Nombre VARCHAR(50) NOT NULL,
    URL VARCHAR(50) NOT NULL
```




```
)TYPE=INNODB;
ALTER TABLE LICENCIATURA ADD FOREIGN KEY (Id_asignatura)
    REFERENCES ASIGNATURA (id)
    ON DELETE CASCADE;
CREATE TABLE LICENCIATURASIGNATURA (
    Id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    id_licenciatura INT NOT NULL,
    id_asignatura INT NOT NULL
)TYPE=INNODB;
ALTER TABLE LICENCIATURASIGNATURA ADD FOREIGN KEY (Id_asignatura)
    REFERENCES ASIGNATURA (id)
    ON DELETE CASCADE;
ALTER TABLE LICENCIATURASIGNATURA ADD FOREIGN KEY (Id_licenciatura)
    REFERENCES LICENCIATURA (id)
    ON DELETE CASCADE;
CREATE TABLE MASTER(
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(50) NOT NULL,
    URL VARCHAR(50) NOT NULL
)TYPE=INNODB;
CREATE TABLE MASTERASIGNATURA (
    Id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    id_master INT NOT NULL,
    id_asignatura INT NOT NULL
)TYPE=INNODB;
ALTER TABLE MASTERASIGNATURA ADD FOREIGN KEY (Id_master)
    REFERENCES MASTER (id)
    ON DELETE CASCADE;
ALTER TABLE MASTERASIGNATURA ADD FOREIGN KEY (Id_asignatura)
    REFERENCES ASIGNATURA (id)
    ON DELETE CASCADE;
CREATE TABLE MASTERLICENCIATURA (
    Id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    id_master INT NOT NULL,
    id_LICENCIATURA INT NOT NULL
) TYPE=INNODB;
ALTER TABLE MASTERLICENCIATURA ADD FOREIGN KEY (Id_master)
    REFERENCES MASTER (id)
    ON DELETE CASCADE;
ALTER TABLE MASTERLICENCIATURA ADD FOREIGN KEY (Id_LICENCIATURA )
    REFERENCES LICENCIATURA (id)
    ON DELETE CASCADE;
CREATE TABLE CURSOSPECIALIZACION(
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(50) NOT NULL,
    URL VARCHAR(50) NOT NULL
)TYPE=INNODB;
CREATE TABLE ESPECIALIZACIONASIGNATURAS (
    Id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
```



```
id_curso INT NOT NULL,
id_asignatura INT NOT NULL

) TYPE=INNODB;
ALTER TABLE ESPECIALIZACIONASIGNATURAS ADD FOREIGN KEY (Id_curso)
REFERENCES CURSOSPECIALIZACION (id)
ON DELETE CASCADE;
ALTER TABLE ESPECIALIZACIONASIGNATURAS ADD FOREIGN KEY (Id_asignatura)
REFERENCES ASIGNATURA (id)
ON DELETE CASCADE;
CREATE TABLE BECA(
id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
Nombre VARCHAR(50) NOT NULL,
Descripcion VARCHAR(50) NOT NULL,
URL VARCHAR(50) NOT NULL,
Titulacion INT NOT NULL,
Cuantia INT NOT NULL

) TYPE=INNODB;
ALTER TABLE BECA ADD FOREIGN KEY (Titulacion)
REFERENCES LICENCIATURA (id)
ON DELETE CASCADE;
CREATE TABLE PFC(
id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
Nombre VARCHAR(50) NOT NULL,
Descripcion VARCHAR(50) NOT NULL,
URL VARCHAR(50) NOT NULL,
Coordinador INT NOT NULL

) TYPE=INNODB;
ALTER TABLE PFC ADD FOREIGN KEY (Coordinador)
REFERENCES USUARIO (id)
ON DELETE CASCADE;
CREATE TABLE TRABAJODIRIGIDO(
id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
Nombre VARCHAR(50) NOT NULL,
Descripcion VARCHAR(50) NOT NULL,
URL VARCHAR(50) NOT NULL,
Coordinador INT NOT NULL

)TYPE=INNODB;
ALTER TABLE PFC ADD FOREIGN KEY (Coordinador)
REFERENCES USUARIO (id)
ON DELETE CASCADE;
CREATE TABLE GRUPOSINVESTIGACION(
id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
Nombre VARCHAR(50) NOT NULL,
Tipo VARCHAR(50) NOT NULL,
Linea_investigacion VARCHAR(50) NOT NULL,
URL VARCHAR(50) NOT NULL

)TYPE=INNODB;
CREATE TABLE INTEGRANTESGRUPOS(
```



```
id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
id_integrante INT NOT NULL,
id_grupo_investigacion INT NOT NULL

)TYPE=INNODB;
ALTER TABLE INTEGRANTESGRUPOS ADD FOREIGN KEY (Id_integrante)
REFERENCES USUARIO(id)
ON DELETE CASCADE;
ALTER TABLE INTEGRANTESGRUPOS ADD FOREIGN KEY (Id_grupo_investigacion)
REFERENCES GRUPOSINVESTIGACION (id)
ON DELETE CASCADE;
CREATE TABLE TESIS(
Titulo VARCHAR(50) NOT NULL PRIMARY KEY,
Id_nombre_autor INT NOT NULL,
Universidad VARCHAR(50) NOT NULL,
URL VARCHAR(50) NOT NULL,
Ano_defensa VARCHAR(50) NOT NULL,
Mes_defensa VARCHAR(50) NOT NULL,
Texto_descriptivo VARCHAR(50) NOT NULL,
departamento VARCHAR(50) NOT NULL

)TYPE=INNODB;
ALTER TABLE TESIS ADD FOREIGN KEY (Id_nombre_autor)
REFERENCES USUARIO(id)
ON DELETE CASCADE;
CREATE TABLE TRANSFERENCIATECNOLOGICA(
id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
Encargo VARCHAR(50) NOT NULL,
Empresa VARCHAR(50) NOT NULL,
Ano_inicio INT NOT NULL,
Ano_fin INT NOT NULL

)TYPE=INNODB;
CREATE TABLE NOTICIA(
id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
nombre VARCHAR(50) NOT NULL,
contenido VARCHAR(50) NOT NULL,
URL VARCHAR(50) NOT NULL,
Fecha DATE NOT NULL

)TYPE=INNODB;
CREATE TABLE AGENDA(
id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
nombre VARCHAR(50) NOT NULL,
URL VARCHAR(50) NOT NULL,
Fecha DATE NOT NULL,
Hora VARCHAR(50) NOT NULL

)TYPE=INNODB;
CREATE TABLE DOCUMENTO(
id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
nombre VARCHAR(50) NOT NULL,
tipo_doc VARCHAR(50) NOT NULL,
```



```
URL VARCHAR(50) NOT NULL,  
Fecha DATE NOT NULL,  
Hora VARCHAR(50) NOT NULL  
  
)TYPE=INNODB;
```



A2 ANEXO PROPUESTA INICIAL AL CLIENTE

Este anexo es el documento íntegro de propuesta inicial que se le hizo al cliente.

A partir de este documento se comenzó a definir los requisitos iniciales que se necesitaban para satisfacer sus necesidades. Las necesidades que se reflejan son incompletas ya que en las reuniones que se realizaron con el cliente se fueron definiendo y ampliando.



1. INTRODUCCI N

Este documento tiene como finalidad definir las funcionalidades b sicas que se pretenden desarrollar en el Proyecto y poder acotar el problema o al menos indicar hasta d nde se va a solucionar. Comenzar  definiendo el objetivo del proyecto que es desarrollar el portal WEB de la Universidad de L rida del departamento de Producci n Vegetal y Ciencia.

Esta aplicaci n se desarrollar  de manera gen rica pudiendo ser usada en cualquier otra universidad u organismo que pretenda almacenar y mostrar informaci n similar.

El proyecto a desarrollar es un Portal WEB de un departamento de una universidad, para ello se utilizar  una Base de Datos similar a la Base de Datos Real del Departamento de Producci n Vegetal y Ciencia.

En este apartado se proceder  a indicar el funcionamiento de la aplicaci n y la arquitectura necesaria para poderse llevar a cabo.

1.1. ALCANCE DEL SOFTWARE

El portal Web del Departamento de Producci n Vegetal y Ciencia de la Universidad de L rida es una aplicaci n Web orientada a gestionar una serie de funcionalidades de consulta y gesti n de la informaci n de un departamento, que pretende ser lo m s sencilla y manejable posible, pero teniendo que cumplir de manera completa con todos los requisitos de funcionamiento solicitados.

En esta aplicaci n Web, cada usuario que la utiliza podr  tener uno o varios permisos y en base a esto, podr  realizar una serie de tareas. Es decir, tendr n tareas limitadas dependiendo del tipo de usuario que sean. Se distinguen tres tipos de usuarios:

- Usuario p blico: usuario que navega por la aplicaci n, pero no requiere registrarse en la aplicaci n.
- Miembro registrado: personal que se registra y pertenece al departamento. Pretende acceder a ciertas funciones que requieren su autenticaci n (en la aplicaci n).



- Administrador del sistema: personal que registrado que puede o no pertenece al departamento (en la aplicación). Pretende acceder a ciertas funciones que requieren su autenticación (en la aplicación).

Dependiendo de si el usuario es administrador o miembro del departamento, tendrán niveles de acceso diferentes en las funciones de gestión (añadir, borrar, modificar o consultar datos) que permite llevar acabo la aplicación.

En el caso de ser un usuario público, tendrá la opción de poder acceder a partes de sólo visualización. Gracias a esta aplicación, un usuario público, podrá acceder a toda la información acerca del departamento, permitiendo conocer cuales son los miembros de dicho departamento, información concreta sobre uno de ellos, así como por ejemplo, la información acerca de los proyectos y publicaciones que han realizado.

A parte de las funcionalidades que han sido solicitadas de consulta y gestión para esta aplicación, la aplicación final tendrá un manual de usuario.

1.2. REQUISITOS DE SOFTWARE NECESARIO

Para el correcto funcionamiento de la aplicación, tendremos que tener instalados los siguientes programas y comprobar que se cumplen los requisitos a continuación citados:

- El Sistema Operativo debe ser Windows.
- Se debe tener instalado e instalar MySql versión 5.0 o superior para almacenar la base de datos que se va a usar en la aplicación.
- Necesitaremos tener instalado Java, con JDK 1.5.
- Se tendrá que instalar, si no se tiene ya instalado, el servidor Apache Tomcat para poder ejecutar la aplicación, ya que se trata de una aplicación Web.

1.3. DEFINICIONES

En este apartado, se pretenden citar y explicar los conceptos que se utilizaran a lo largo del documento y que pueden causar dudas o confusión al lector.



- ◆ JAVA: Lenguaje de programación orientado a objetos.
- ◆ TOMCAT: Servidor Web que soporta aplicaciones web dinámicas basadas en servlets y jsp.
- ◆ MYECLIPSE: Entorno integrado de desarrollo de aplicaciones JAVA.
- ◆ MYSQL: Sistema gestor de bases de datos.
- ◆ JSP: Tecnología Java que permite el desarrollo software para generar documentos en respuesta a peticiones de un cliente web.
- ◆ SERVLETS: Objeto que recibe una petición y genera una respuesta basada en esta.



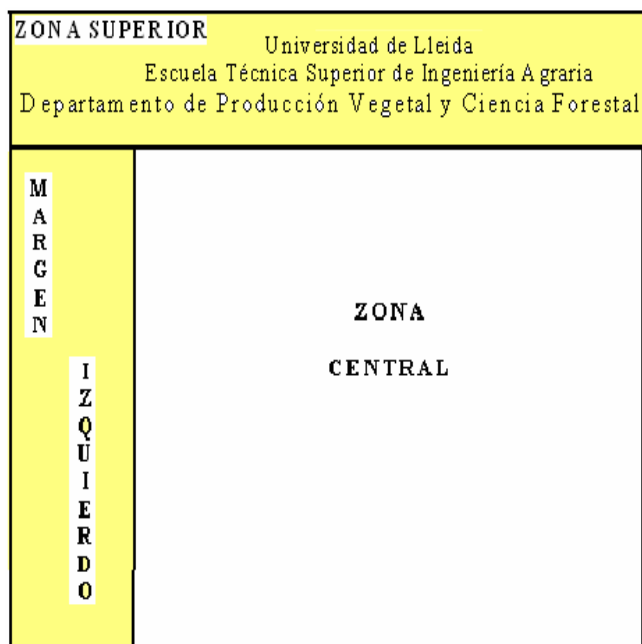
2. OBJETIVOS DEL SISTEMA A DESARROLLAR

Tras la pequeña introducción anterior, tanto al problema como al documento, en esta segunda sección, se defina las funcionalidades de la aplicación Web a desarrollar.

2.1. ESTRUCTURA GENERAL

Los objetivos se han dividido en tres grandes grupos para facilitar su visión, separando así las diferentes partes de que constará el sistema:

La página principal se dividirá en 3 zonas: zona superior, margen izquierdo, zona central. En la siguiente imagen se puede observar cual será la organización de cada zona.



- ❖ La **zona superior** será fija en toda las páginas y mostrará el logo y nombre del departamento, similar a



Universidad de Lleida

Escuela Técnica Superior de Ingeniería Agraria

Departamento de Producción Vegetal y Ciencia Forestal

❖ En el **margen izquierdo** se mostrará un menú fijo con las operaciones a realizar que son:

- *¿Quién somos?*
- *Personal*: desplegable con : PDI (Personal docente investigador), PAS (Personal de administración y servicios) y Becarios
- *Docencia*: desplegable con: Docencia de Grado, Máster, Cursos especialización
- *Investigación*: Grupos de investigación, Tesis doctorales, transferencia de tecnología
- *Ofertas*: desplegable con: Becas, Proyectos Final de Carrera, Trabajos
- *Acceso restringido* mediante: Nombre de *usuario* y *password*.

❖ En la **zona central** aparecerán información general del departamento. Inicialmente aparecerá una agenda y un conjunto de noticias. Cuando se seleccione algún link del margen izquierdo, aparecerá información referente a la petición solicitada. Empezando con una *Cabecera* que alude a la opción seleccionada previamente.

Las diferentes secciones del sitio a las que vaya accediendo el usuario se mostraran en la zona central de la página, sin que sufran modificaciones las otras tres zonas.

2.2. TIPOS DE USUARIOS

Existirán dos tipos de usuarios del sitio: usuarios *registrados* y usuarios *no registrados*.

❖ **Usuario no registrados**

- Los usuarios no registrados del sistema sólo podrán visualizar los resultados de las siguientes consultas:
 - i. Noticias y agenda de la pantalla principal.
 - ii. ¿Quiénes somos?



1. Seleccionar miembros del departamento y poder ver sus curriculums.
- iii. Docencia
 1. Docencia de Grado
 2. Masters
 3. Cursos de especialización
- iv. Investigación
 1. Grupos de investigación
 2. Tesis doctorales
 3. Transferencia de tecnología
- v. Ofertas
 1. Becas
 2. Proyectos de Final de Carrera
 3. Trabajos dirigidos



❖ **Usuario registrado:** Los usuarios registrados deben estar identificados en una Base de Datos como miembros del Departamento o como usuarios administrador. Las operaciones en líneas generales serán:

- Después de “loguearse”, se procede a ver una página que contendrá información propia del departamento:
 - i. Noticias
 - ii. Agenda
- Reservas: los miembros del departamento podrán realizar reservas de: coches, laboratorios, ordenadores y cañones.
- Añadir , borrar o modificar noticias
- Añadir , borrar o modificar datos de la agenda.

2.3. PAGINA PRINCIPAL

La página principal contiene *información pública* que se muestra a cualquier usuario sin necesidad de que esté registrado en la aplicación y permite la opción de *acceso a zonas restringidas*, si es un usuario autorizado (administrador o miembro del departamento).

La información se mostrará de manera similar a la indicada a continuación:

Universidad de Lleida Escuela Técnica Superior de Ingeniería Agraria Departamento de Producción Vegetal y Ciencia Forestal					
	[catalán] [] []				
<ul style="list-style-type: none">• ¿Quién somos?• Personal• Docencia• Investigación• Ofertas	<table border="1"><tr><td>AGENDA</td><td>NOTICIA1</td></tr><tr><td>NOTICIA3</td><td>NOTICIA2</td></tr></table>	AGENDA	NOTICIA1	NOTICIA3	NOTICIA2
AGENDA	NOTICIA1				
NOTICIA3	NOTICIA2				
<table border="1"><tr><td>Usuario</td><td><input type="text"/></td></tr><tr><td>Password</td><td><input type="password"/></td></tr></table>	Usuario	<input type="text"/>	Password	<input type="password"/>	
Usuario	<input type="text"/>				
Password	<input type="password"/>				

PÚBLICO (indicated by a red line pointing to the top section)

PRIVADO (indicated by a blue line pointing to the bottom section)

Referente a la información anterior se comenta:

- Se da la opción de seleccionar el idioma en el que se quiere mostrar la información, los posibles son: castellano, catalán e inglés.
- ¿Quiénes somos?: Información descriptiva del departamento, miembros, actividades a las que se dedican, etc.
- Personal contendrá un desplegable con : PDI (Personal docente investigador), PAS (Personal de administración y servicios) y Becarios

- Docencia contendrá un desplegable con: Docencia de Grado, Máster, Cursos especialización
- Investigación: Grupos de investigación, Tesis doctorales, transferencia de tecnología
- Ofertas contendrá un desplegable con: Becas, Proyectos Final de Carrera, Trabajos.
- La agenda constará de los campos: días, mes, hora y descripción.

Ejemplo de agenda:

Dilluns 03/03/08 11.00h	Presentació de les XIX Jornades Universitat-Empresa. Qualitat i Gestió Econòmica , que se celebren els dies 3 i 4 de març AUDITORI DEL CENTRE DE CULTURES I COOPERACIÓ TRANSFRONTERERA (CAMPUS DE CAPPONT)	Dilluns 03/03/08 20.00h	Retransmissió de l'òpera Elektra , de Richard Strauss, dins el programa Opera Oberta 2007/2008. El Liceu a la UdL. SALA D'ACTES (RECTORAT)
--	--	--	---

- Las noticias contendrán los campos: título, fecha y descripción. Ejemplo de noticia.

	<p>título</p> <p>La UdL invertirà 3 milions en 3 anys per adequar els seus espais a l'EEES</p> <p>fecha</p> <p>03/03/08/</p> <p>descripció</p> <p>La Universitat de Lleida invertirà 3 milions d'euros en 3 anys per adequar els seus espais a l'Espai Europeu d'Educació Superior. El Consell de Govern, entre d'altres acords, ha donat llum verda avui a l'inici d'aquest procés que suposarà la remodelació i adequació d'aules i seminaris, així com la compra d'equipament informàtic, de mobiliari, o l'ampliació de les zones wifi. [+]</p>
--	--



2.4. INFORMACIÓN PÚBLICA

Esta información se corresponde con la información que proporciona la página inicial y las subpáginas a las que se puede acceder desde la misma sin tener que registrarse. A continuación se describe la información que mostrarán las pantallas a las que se accede en función de la petición solicitada:

2.4.1. ¿QUIENES SOMOS?

Universidad de Lleida Escuela Técnica Superior de Ingeniería Agraria Departamento de Producción Vegetal y Ciencia Forestal					
[catalán] [es] [en]					
<ul style="list-style-type: none">• ¿Quién somos?• Personal• Docencia• Investigación• Ofertas <p>Usuario <input type="text"/></p> <p>Password <input type="password"/></p>	<table border="1"><tr><td>AGENDA</td><td>NOTICIA1</td></tr><tr><td>NOTICIA3</td><td>NOTICIA2</td></tr></table>	AGENDA	NOTICIA1	NOTICIA3	NOTICIA2
AGENDA	NOTICIA1				
NOTICIA3	NOTICIA2				

Universidad de Lleida Escuela Técnica Superior de Ingeniería Agraria Departamento de Producción Vegetal y Ciencia Forestal	
<ul style="list-style-type: none"> • ¿Quién somos? • Personal • Docencia • Investigación • Ofertas 	¿Quién somos? [catalán] [] []
	Texto descriptivo
	Foto: Personal Dpto.
	Enlace con un video de imágenes del Departamento
Usuario <input type="text"/> Password <input type="password"/>	

Como se puede observar se mantendrá el margen izquierdo durante toda la navegación de páginas, de tal forma que siempre se pueda acceder a información no restringida sin necesidad de tener que volver a la página principal de la aplicación. La información que se mostrará es:

- Cabecera de la opción selecciona del margen izquierdo: ¿Quiénes somos?
- Texto que describe el departamento, como por ejemplo: : (texto en catalán)

Respecte a la producció vegetal, el Departament està especialitzat en agronomia, biologia, biotecnologia agrària, control integrat de plagues, ecologia agrícola, genètica, millora genètica vegetal i patologia vegetal. Quant al vessant de tecnologia forestal, s'especialitza en focs forestals, ecologia forestal, repoblació forestal, silvicultura, pascicultura, genètica forestal, entomologia forestal i patologia forestal. A més a més, hi ha professorat del Departament que està especialitzat en disseny d'experiments i anàlisi de dades en l'àmbit de la investigació agrícola i forestal.

El Departament està format per 44 professors e investigadores y 6 personas de administración y servicios. En la actualidad además lo integran 13 becarios de doctorado.





- Foto del departamento
- Enlace a un video del departamento.

2.4.2. PERSONAL

En el margen izquierdo se podrá seleccionar qué tipo de personal se quiere visualizar, esto se conseguirá mediante un desplegable con las opciones de:

- PDI: Personal docente de Investigación.
- PAS: Personal de administración y Servicios.
- Becarios

Independientemente de la opción señalada, la página resultante tendrá un formato similar a la figura siguiente.

Universitat de Lleida Escuela Técnica Superior de Ingeniería Agraria Departamento de Producción Vegetal y Ciencia Forestal					
<ul style="list-style-type: none">• ¿Quién somos?• Personal• Docencia• Investigación• Ofertas <p>Usuario <input type="text"/></p> <p>Password <input type="password"/></p>	<p>[catalán] [] []</p> <table border="1"><tr><td>AGENDA</td><td>NOTICIA1</td></tr><tr><td>NOTICIA3</td><td>NOTICIA2</td></tr></table>	AGENDA	NOTICIA1	NOTICIA3	NOTICIA2
AGENDA	NOTICIA1				
NOTICIA3	NOTICIA2				

Dependiendo de del tipo de personal seleccionado en el margen izquierdo, se mostrará información de distintos miembros.



Universitat de Lleida Escuela Técnica Superior de Ingeniería Agraria Departament de Producció Vegetal y Ciencia Forestal	
<ul style="list-style-type: none">• ¿Quién somos?• Personal• Docencia• Investigación• Ofertas Usuario <input type="text"/> Password <input type="text"/>	Personal [catalán] [] [] <div>Texto descriptivo</div> <div>Datos Miembro Cargo Datos Miembro Cargo Datos Miembro Cargo Datos Miembro Cargo Datos Miembro Cargo Datos Miembro Cargo Datos Miembro Cargo Datos Miembro Cargo</div>

Los datos de un miembro mostrado pueden ser:

Apellido1 Apellido2, Nombre.

Y respecto al cargo que ocupan dentro de la universidad puede ser:

- Si es un personal del PDI:
 - Catedrático
 - Titular
 - Profesor ayudante
- Si es un PAS:
- Si es un Becario



2.4.3. PÁGINA PERSONAL

Si se selecciona algún miembro del departamento, se mostrará el currículum de la persona. Cada miembro tiene ya generada su página Web con su información personal. Así que el portal Web sólo tendrá que almacenar esta información por cada persona.

2.4.4. DOCENCIA

En esta operación se selecciona una de las opciones que aparecen en el desplegable. La cual puede ser:

- Docencia de Grado: Como resultado de la ejecución mostrará la información de las titulaciones y sus correspondientes asignaturas. Por cada titulación, aparecerán “n” asignaturas y cada una de ellas tendrá:
 - Nombre de la Asignatura
 - “url” con una información “pdf”
 - dossier-e: www.paginaWebDelaAsignatura.es
- Master
 - Nombre del master con un enlace al sitio de interés.
- Cursos de especialización
 - Nombre del curso con un enlace al sitio de interés.

2.4.5. OFERTAS

En esta operación se selecciona una de las opciones que aparecen en el desplegable. La cual puede ser:



- Becas
- Proyectos Finales de carrera
- Trabajos Dirigidos

Cuando se selecciona una de estas opciones, muestra en la pantalla la página Web que contiene la información.

2.4.6. INVESTIGACIÓN

El resultado de la operación muestra una salida similar a esta, compuesta por:

- Texto descriptivo
- Las tres posibles vías de investigación, las cuales tienen un enlace a un sitio Web.

Universitat de Lleida Escuela Técnica Superior de Ingeniería Agraria Departamento de Producción Vegetal y Ciencia Forestal	
	Investigación [catalán] [] []
<ul style="list-style-type: none">• ¿Quién somos?• Personal• Docencia• Investigación• Ofertas	<div>Texto descriptivo</div> <div>Grupos de investigación Tesis doctorales Transferencia de tecnología</div>
Usuario <input type="text"/>	
Password <input type="text"/>	

Si seleccionamos:

- [Grupos de investigación](#):
 - Grupos consolidados por la Generalitat, con sus correspondientes *enlaces*.
 - Grupos consolidados por la UdL, con sus correspondientes *enlaces*.
En los *enlaces* se mostrará los *integrantes* que componen ese grupo y sus *líneas de investigación*.
- [Tesis Doctorales](#)
 - Texto
 - Relación Tesis



- Transferencia Tecnológica
 - Texto

2.5. INFORMACIÓN RESTRINGIDA

Contendrá información reservada a los miembros del Departamento. Por este motivo se necesita introducir un *usuario* y *password* que muestre información dependiendo del tipo de usuario que sea.

Universitat de Lleida Escola Tècnica Superior d'Enginyeria Agrària Departament de Producció Vegetal i Ciència Forestal					
<ul style="list-style-type: none">• ¿Quiéu somos?• Personal• Docencia• Investigación• Ofertas	<table border="1"><tr><td>AGENDA</td><td>NOTICIA1</td></tr><tr><td>NOTICIA3</td><td>NOTICIA2</td></tr></table>	AGENDA	NOTICIA1	NOTICIA3	NOTICIA2
AGENDA	NOTICIA1				
NOTICIA3	NOTICIA2				
<div>Usuario <input type="text"/></div> <div>Password <input type="password"/></div>					

La información que contendrá este sitio de acceso restringido, será:

- Documentos para poder descargarse, se podrá mostrar como un desplegable con:
 - Estatutos
 - Actas
 - Consejos Departamentos
 - Información económica
 - Docencia
 - Investigación
 - Convocatorias
- Agenda.
- Noticias.
- Permitirá realizar reservas:
 - De medios materiales: coches, ordenadores, cañones.



- Laboratorios del Dpto.

La página principal contendrá la siguiente información:

